Lehrstuhl für Solartechnik

RWTH AACHEN UNIVERSITY

The present work was submitted to the Institute of Solar Research of
German Aerospace Center
(Deutsches Zentrum für Luft- und Raumfahrt)

Evaluation and Optimization of Heliostat Calibration Methods

Bachelor's thesis
of
Felix Zimmermann,
Matriculation Number 366556

Field of study:	Computational Engineering Science
Duration of this work:	12 Weeks
Submitted on:	26.05.2021
Supervisor:	Dr.-Ing. Daniel Maldonado Quinto

Ich versichere, diese Arbeit im Rahmen der am Institut üblichen Betreuung selbständig angefertigt und keine anderen als die angegebenen Quellen verwendet zu haben.

(Felix Zimmermann)

Aachen, 26.05.2021

Abstract

In solar tower power plants, the sun's energy is focused onto a receiver at the top of a solar tower by means of many biaxially movable mirrors (heliostats) and the high energy density generated is used, for example, to generate electricity via a steam power process or to trigger chemical processes for the production of new types of fuels. It is very important that the heliostats track the movement of the sun correctly throughout the day, so that the incoming solar rays are reflected as accurately as possible onto a defined point on the tower. To achieve this, motor positions of the two linear actuators, which change the orientation of the heliostats, are calculated from an error-prone kinematic geometry model of the individual heliostats. To minimize alignment errors, selected parameters of the geometry model are adjusted by means of a regression to calibration data recorded by the camera-target method. Within the scope of this work, on the one hand, the regression method used is investigated and improvements and further algorithms are tested and developed. On the other hand, the selection of optimization parameters from the geometry model is analyzed. It was found that changing the regression methodology does not improve the alignment accuracy. However, a selection of optimization parameters and their optimization order could be found that indicate a significant reduction of the alignment error.

Kurzfassung

In Solarturmkraftwerken wird die Energie der Sonne mittels vieler zweiachsig bewegbaren Spiegel (Heliostate) auf einen Receiver an der Spitze einen Solarturmes fokussiert und die dabei erzeugte hohe Energiedichte genutzt um beispielsweise Strom über einen Dampfkraftprozess zu erzeugen oder chemische Prozesse zur Erzeugung neuartiger Treibstoffe anzustoßen. Dabei ist es von sehr großer Bedeutung, dass die Heliostate der Bewegung der Sonne über den Tag korrekt nachgeführt werden, sodass die eintretenden Sonnenstrahlen möglichst genau auf einen definierten Punkt am Turm reflektiert werden. Dafür werden aus einem fehlerbehafteten kinematischen Geometriemodell der einzelnen Heliostate Motorpositionen der beiden Linearaktuatoren berechnet, welche die Orientierung der Heliostate verändert. Um Ausrichtungsfehler zu minimieren, werden ausgewählte Parameter des Geometriemodells mittels einer Regression an Kalibrierdaten, die mittels der Kamera-Target-Methode aufgezeichnet werden, angepasst. Im Rahmen dieser Arbeit wird zum einen das verwendete Regressionsverfahren untersucht sowie Verbesserungen und weitere Algorithmen getestet und entwickelt. Zum anderen wird die Auswahl der Optimierungsparameter aus dem Geometriemodell analysiert. Es wurde festgestellt, dass die Veränderung der Regressionsmethodik keine Verbesserung der Ausrichtungsgenauigkeit mit sich bringt. Jedoch konnte eine Auswahl an Optimierungsparametern und deren Optimierungsreihenfolge gefunden werden, die eine erhebliche Reduzierung des Ausrichtungsfehlers andeutet.

Contents

List of Figures

List of Tables

List of Symbols

F Target function of optimization

M Geometry model of a heliostat

P Pulse ratio of heliostat driving actuator

Q Quadratic model of target function

X Motor position of heliostat driving actuator

Δ Step size in the difference quotient

Ω Trust Region

α Basic tilt of primary rotation axis of heliostat regarding ENU-COS

β Basic tilt of secondary rotation axis of heliostat regarding ENU-COS

δ Mirror tilt in relation to secondary rotation axis of heliostat

γ Tilt of secondary rotation axis in relation to primary rotation axis of heliostat

λ Damping parameter in the Levenberg-Marquardt algorithm

$\mathbf{I}$ Identity matrix

$\mathbf{J}$ Jacobian matrix of residuals

$\mathbf{f}$ General set of residuals

$\mathbf{g}$ Gradient of target function

$\mathbf{h}$ Parameter step in optimization algorithm

$\mathbf{p}$ Set of model fitting parameters

$\mathbf{x}$ General set of variables

$\overline{\overline{R}}_{01}$ Rotation matrix describing alignment of first rotation axis of heliostat regarding ENU-COS

$\overline{\overline{R}}_{12}$ Matrix describing rotation from H-COS to F-COS

$\overline{\overline{R}}_{2M}$ Matrix describing rotation from F-COS to G-COS

$\overline{\overline{R}}_\tau$ Matrix describing rotation around secondary rotation axis of heliostat

$\overline{\overline{R}}_\theta$ Matrix describing rotation around primary rotation axis of heliostat

$\overline{\overline{R}}_x$ Matrix performing rotation around x-axis

$\overline{\overline{R}}_y$ Matrix performing rotation around y-axis

$\overline{\overline{R}}_z$ Matrix performing rotation around z-axis

ρ Gain ratio in the Levenberg-Marquardt algorithm, i.e. the ratio between the actual and predicted decrease in function value

τ_K Fixed angular offset of tilt around secondary rotation axis of heliostat

τ_{max} Maximal possible tilt around second rotation axis of heliostat

τ_{min} Minimal possible tilt around second rotation axis of heliostat

τ Tilt around secondy rotation axis of heliostat

θ_K Fixed angular offset of tilt around primary rotation axis of heliostat

θ_{max} Maximal possible tilt around first rotation axis of heliostat

θ_{min} Minimal possible tilt around first rotation axis of heliostat

θ Tilt around primary rotation axis of heliostat

φ Traverse angle of heliostat driving geometry

ϑ Tracking error, defined as the angle between the ideal and actual normal vector

$\vec{a}$ Vector describing the aim point in ENU-COS

$\vec{s}$ Vector describing sun direction in the ENU-COS

$\vec{h}$ Position of H-Origin at first rotation axis of heliostat in the ENU-COS

$\vec{m}$ Mirror center position of heliostat in ENU-COS

$\vec{n}$ Normal vector of heliostat mirror surface in ENU-COS

$\vec{t}$ Position of calibration target origin in the ENU-COS

$\vec{v}'_{12}$ $\vec{v}_{12}$ transformed into ENU-COS

$\vec{v}_{12}$ Vector in the H-COS describing the shortest path between first and second rotation axis of heliostat

$\vec{v}'_{2M}$ $\vec{v}_{2M}$ transformed into ENU-COS

$\vec{v}_{2M}$ Vector in the F-COS describing the shortest path between second rotation axis and mirror center of heliostat

$\vec{x}_2$ Secondary rotation axis of heliostat

$\vec{y}_3$ Axis in G-COS describing the normal vector of mirror surface

$\vec{z}_1$ Primary rotation axis of heliostat

a Kinematic parameter of heliostat

b Kinematic parameter of heliostat

c Kinematic parameter of heliostat

d Kinematic parameter of heliostat

e Kinematic parameter of heliostat

f Single residual value

h_e East-Component of the H-Origin position

h_n North-Component of the H-Origin position

h_u Up-Component of the H-Origin position

v_{12} Smallest distance between first and second rotation axis of heliostat

v_{2M} Shortest distance between second rotation axis and mirror center of heliostat

x Stroke length of heliostat driving geometry

Acronyms

CSP Concentrated Solar Power 1

DL Dog Leg 43

DLR German Aerospace Center 2, 52

DNI Direct Normal Irradiance 19

ENU-COS East-North-Up-Coordinate-System 7, 8, 10, 16, 23, 31

F-COS F-Coordinate-System 7, 8

G-COS G-Coordinate-System 8

GN Gauss-Newton 21

GPS Global Positioning System 12

GUI Graphical User Interface 12

H-COS H-Coordinate-System 7, 8

LM Levenberg-Marquardt 18, 21, 22, 33, 35, 39–43, 45

MSE Mean Squared Error 20

PV Photovoltaics 1

Redox reduction-oxidation 5

SOTA State-of-the-Art 45, 48

SPA Sun Position Algorithm 12

SQL Structured Query Language 12

STJ Jülich Experimental Solar Thermal Power Plant 1–3, 5

TRF Trust Region Reflective 43

Glossary

CET Central European Time 23, 30

F-Origin Origin of the F-Coordinate-System 7, 8

G-Origin Origin of the G-Coordinate-System 8

H-Origin Origin of the H-Coordinate-System 7

MINPACK A library of FORTRAN subroutines for the solving of systems of nonlinear equations, or the least squares minimization of the residual of a set of linear or nonlinear equations 42

UTC Coordinated Universal Time 23

1 Introduction

In terms of global energy supply, our society is facing the challenges of global warming due to CO2 emissions and the depletion of fossil raw material reserves. At the same time, the demand for energy is expected to continually rise due to the increase in living standards in many areas of the world. [11]

To meet these challenges, many countries are already making increased use of renewable energies. Various technologies from the field of renewable energies use the sun as an unlimited source of energy. Photovoltaics (PV), in which solar radiation is converted directly into electrical energy via the photoelectric effect, is widely used. Another possibility is the use of the thermal energy of solar radiation in concentrated form, so-called Concentrated Solar Power (CSP). This can be used to drive conventional power plants and chemical engineering processes. Due to the cost-effective and efficient intermediate heat storage, CSP offers the significant advantage of being able to provide energy on a time-delayed and demand-driven basis, for example at night. [18]

One type of CSP plants, that is currently the subject of extensive research activities, are solar power tower plants. At solar power tower plants, the solar energy is concentrated with the help of many mirrors (heliostats) onto a radiation receiver, which is installed at the top of a tower and in which the concentrated solar radiation from the heliostat field is converted into thermal energy. Using this technology it is possible to provide heat transfer medium temperatures up to 1400 °C and the provision of electrical power, thermal process energy and fuel synthesis have already been successfully tested. [21, 10, 16, 17]

Figure 1.1 shows a view from the field of heliostats towards the solar tower at the Jülich Experimental Solar Thermal Power Plant (STJ) in operation. More than 2000 biaxially adjustable heliostats are installed here. New radiation receivers for a more efficient use of the incident radiation and also technologies for synthetic fuel synthesis are tested. In addition, the solar energy can be used to generate up to 1.5 MW of electricity in a downstream steam power process.

Figure 1.1: View from field of heliostats towards solar tower at the Jülich Experimental Solar Thermal Power Plant in operation [6]

In Jülich, the number of annual solar hours is about 1500 hours (1496.5 hours in Cologne [2]) which is a lot less than in more southern places, e.g Alméria (Spain) with 3066 hours per year [1]. From this point of view and the fact that the alignment of the heliostats directly affects the amount of electricity that can be generated in the power plant, it is very important that the tracking of the heliostats works perfectly, so that the sunlight is always reflected to the desired target point and thus the available solar energy can be used efficiently. To ensure this, the actual orientation of each heliostat is recorded in regular sections and compared with the desired orientation. For this purpose, the so-called Camera-Target-Method (also called Stone-Method [19]) is used at the STJ for the calibration of the heliostats. In this process, the focal point of each heliostat is shifted individually to a calibration target, which is located underneath the receiver. A camera then takes a picture of the calibration target and the position of the focal spot's centroid is determined and compared to the desired aiming point on the target. After storing all information in a database, an error-based geometry is fitted by regression to represent the real heliostat behaviour as best as possible. Due to the number of heliostats and limited sun hours, a maximum of around 20 calibration images per heliostat can be collected over the year under regular operating conditions.

The aim of the thesis is to evaluate the calibration method, implemented in the heliostat controlling software HeliOS, developed by the German Aerospace Center (DLR), and to work out optimization measures to improve the tracking accuracy of heliostats using as few

calibration data as possible.

Chapter 2 first provides the basics knowledge for understanding the work and includes the current state of the art of the implemented calibration procedure at the STJ. The main part of this thesis starts in chapter 3, where first a sensitivity analysis of selected parameters of the underlying geometry model of the heliostats is performed, the regression method used so far is investigated and afterwards different improvement measures are identified, using simulated calibration data. Then, using the results from chapter 3 derived methods for calibration are tested with real calibration data collected at the STJ. Finally, a summary conclusion is given in chapter 5, followed by an outlook for further approaches to heliostat calibration methods, in which the results obtained in this thesis can have an influence.

2 Principles

2.1 Basic Concepts of Solar Power Tower Plants

In a solar power tower plant the incident solar radiation is focused on a small point at the head of a tower and therefore it is assigned to the point-focusing solar systems. Other solar systems for example include line-focusing systems, such as parabolic trough power plants. [18]

A solar power tower plant consists of a tower and several adjustable mirrors surrounding the tower, called heliostats. Regarding the current position of the sun, every heliostat is adjusted such that the sun-rays are reflected to a defined point on the tower. The heliostats are adjustable by two actuators, which can move the heliostat and track the moving sun biaxially. At the point, where the radiation of the sun is focused, there is positioned a receiver, which can use the heat for different industrial tasks, e.g. the generation of electricity in a steam power process or the creation of synthetic fuels in chemical reduction-oxidation (Redox) reactions. Figure 2.1 shows the basic concept of a solar power tower plant, including a conventional steam power plant process (Rankine cycle). The thermal energy collected at the receiver is used to heat a heat transfer medium, such as air or liquid nitrate. The heat transfer medium conducts the thermal energy into a heat exchanger, where the energy is used to evaporate water and superheat it up to 600 °C. By expanding the water vapor in a turbine, a generator is operated via a shaft, which converts the mechanical energy into electrical current and feeds this into the power grid. In addition, a part of the thermal energy collected at the solar tower is stored in a heat storage to keep the steam power process going even at times when the sun is not shining, such as at night. So-called latent heat storages are used, where the temperature of the heat storage medium, mostly special type of nitrate, does not change, but the aggregate state is changed. The melting of the heat storage medium stores energy, while the solidification releases this energy again [18]. In order to counteract fluctuations in the sun's output throughout the day, the heat transfer medium in the STJ receiver is not heated directly by the sun's energy, but first heats a heat accumulator made of ceramic fill, through which the air to be heated is then passed and heated [5].

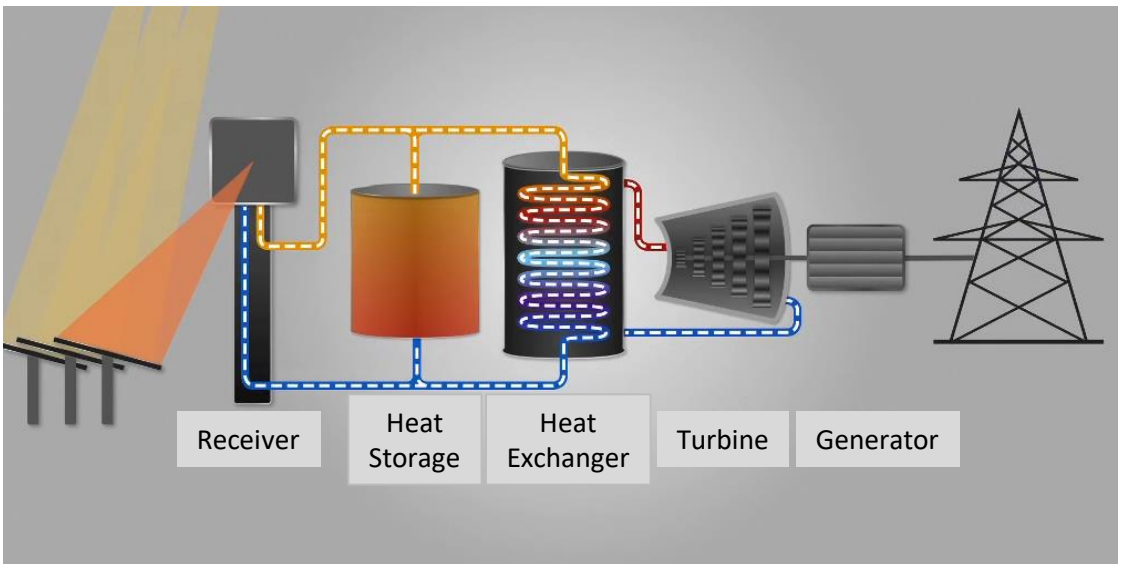


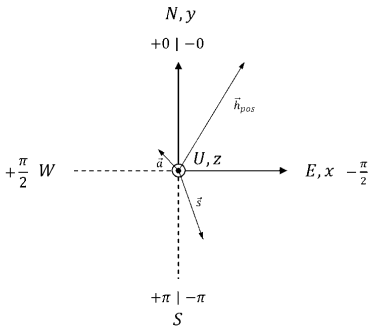
Figure 2.1: Basic concept of solar power tower plant [4]

One basic challenge during the solar operation is to track the heliostat orientation regarding the variable sun position, such that the reflection actually hits the receiver. Besides the calculation of the exact course of the sun position, considering the location of the power plant, detailed information about each heliostat is needed. This information includes the heliostat position with reference to the tower, geometric dimensions and knowledge about the lever arms of the actuators. For each heliostat a geometric model can be derived from this information, which can be used to calculate the needed motor positions of the actuators, such that the desired reflection point for a given sun position is hit. The geometric model is described in the next section 2.2. A method to get an idea about the precision of the tracking, and which can also be used to calibrate heliostats, is the Camera-Target-Method. The calibration system using the Camera-Target-Method and the calibration algorithm at the solar tower plant in Jülich are described in sections 2.5 and 2.6.

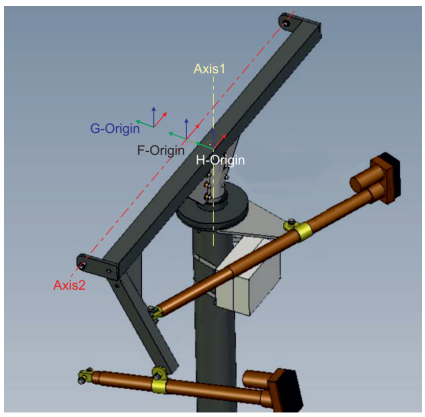
2.2 Geometric Model of Heliostat

To calculate the motor positions of the two actuators, which are needed to align a heliostat, such that the sun-rays are reflected to the desired aim point, a mathematical model of the heliostats' geometry and kinematic is essential. First the geometric description of the heliostat is defined, before the calculation routine of motor positions is explained.

The geometric model of a heliostat describes the kinematics of the heliostats mirror surface



(a) Global coordinate system



(b) Axes and local coordinate systems of heliostat

Figure 2.2: Global and local coordinate systems

as it rotates about two axes that are locally offset from and perpendicular to each other. It is derived by three interdependent coordinate systems, shown in figure 2.2b. The origin of the H-Coordinate-System (H-COS) is located at the point on the first rotation axis with the smallest distance v_{12} to the second rotation axis of the heliostat, such that the z-axis is parallel to the first rotation axis and the y-axis points to the second rotation axis. The position of the H-Origin in the global coordinate system is given by $\vec{h}$. The global coordinate system has its origin at the foot of the solar tower, with the x-axis orientated to the east, y-axis to the north and z-axis upwards, and it is called East-North-Up-Coordinate-System (ENU-COS) (see figure 2.2a). The H-COS is twisted compared to the ENU-COS by the angle α around the z-axis and the angle β around the x-axis, and this alignment can be described by the rotation matrix $\overline{\overline{R}}_{01}$ (see equation 2.1).

$$\overline{\overline{R}}_{01} = \overline{\overline{R}}_x(\beta) \cdot \overline{\overline{R}}_z(\alpha) \tag{2.1}$$

The rotation axis of H-COS is called $\vec{z}_1$, being the first rotation axis of the heliostat, with rotation angle (θ+θ_K), where θ is the variable tracking angle and θ_K a fixed offset. Let this rotation be defined by the rotation matrix $\overline{\overline{R}}_\theta$ (see equation 2.2).

$$\overline{\overline{R}}_\theta = \overline{\overline{R}}_z(\theta + \theta_K) \tag{2.2}$$

Vector $\vec{v}_{12}$, connecting the H-Origin and F-Origin, the origin of the F-Coordinate-System (F-COS), is defined in the H-COS as in equation 2.3.

$$\vec{v}_{12} = v_{12} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tag{2.3}$$

To describe vector $\vec{v}_{12}$ in the ENU-COS it has to be rotated backwards with negative angle values used in $\overline{\overline{R}}_\theta$ and $\overline{\overline{R}}_{01}$ (see equation 2.4).

$$\begin{aligned}\vec{v}'_{12} &= \overline{\overline{R}}_z(-\alpha) \cdot \overline{\overline{R}}_x(-\beta) \cdot \overline{\overline{R}}_z(-(\theta+\theta_K)) \cdot \vec{v}_{12} \\ &= \overline{\overline{R}}_{01}{}^T \cdot \overline{\overline{R}}_\theta{}^T \cdot \vec{v}_{12}\end{aligned} \tag{2.4}$$

The second rotation axis, which is the x-axis in the F-COS and is called $\vec{x}_2$, may not be perpendicular to the first rotation axis $\vec{z}_1$ and therefore the F-COS is rotated by an angle γ around the y-axis in relation to the H-COS. The rotation around $\vec{x}_2$ is given by the angle (τ+τ_K), with τ beeing the tracking angle of the elevation axis and τ_K a fixed offset. The F-Origin is located at the point on the second rotation axis, with the smallest distance v_{2M} to the mirror center, such that the F-Origin and G-Origin are connected by vector $\vec{v}_{2M}$, given in the F-COS as in equation 2.5.

$$\vec{v}_{2M} = v_{2M} \cdot \begin{pmatrix}0\\1\\0\end{pmatrix} \tag{2.5}$$

Defining $\overline{\overline{R}}_{12}$ and $\overline{\overline{R}}_\tau$ as in equations 2.6 and 2.7, $\vec{v}_{2M}$ can be described in the ENU-COS according to equation 2.8.

$$\overline{\overline{R}}_{12} = \overline{\overline{R}}_y(\gamma) \tag{2.6}$$

$$\overline{\overline{R}}_\tau = \overline{\overline{R}}_x(\tau+\tau_K) \tag{2.7}$$

$$\begin{aligned}\vec{v}'_{2M} &= \overline{\overline{R}}_z(-\alpha) \cdot \overline{\overline{R}}_x(-\beta) \cdot \overline{\overline{R}}_z(-(\theta+\theta_K)) \cdot \overline{\overline{R}}_y(-\gamma) \cdot \overline{\overline{R}}_x(-(\tau+\tau_K)) \cdot \vec{v}_{2M} \\ &= \overline{\overline{R}}_{01}{}^T \cdot \overline{\overline{R}}_\theta{}^T \cdot \overline{\overline{R}}_{12}{}^T \cdot \overline{\overline{R}}_\tau{}^T \cdot \vec{v}_{2M}\end{aligned} \tag{2.8}$$

From this the mirror center position $\vec{m}$ in the ENU-COS is defined by equation 2.9.

$$\vec{m} = \vec{h} + \vec{v}'_{12} + \vec{v}'_{2M} \tag{2.9}$$

The y-axis of the G-Coordinate-System (G-COS) is called $\vec{y}_3$ and represents the normal vector $\vec{n}$ of the mirror. A tilt of the mirror compared to the F-COS is modeled by a rotation around the z-axis with the angle δ and is represented by the rotation matrix $\overline{\overline{R}}_{2M}$, defined in equation 2.10.

$$\overline{\overline{R}}_{2M} = \overline{\overline{R}}_z(\delta) \tag{2.10}$$

Given the tracking angles θ and τ, and the geometry parameters α, β, γ, δ, θ_K and τ_K, the normal vector $\vec{n}$ is given in the ENU-COS by equation 2.11.

$$\vec{n} = \overline{\overline{R}}_{01}{}^T \cdot \overline{\overline{R}}_\theta{}^T \cdot \overline{\overline{R}}_{12}{}^T \cdot \overline{\overline{R}}_\tau{}^T \cdot \overline{\overline{R}}_{2M}{}^T \cdot \begin{pmatrix}0\\1\\0\end{pmatrix} = \begin{pmatrix}n_e\\n_n\\n_u\end{pmatrix} \tag{2.11}$$

To get the tracking angles for a desired normal vector, equation 2.11 is rearranged. First the u-component of $\vec{n}$ can be used to determine the rotation angle of the elevation rotation axis (see equation 2.12).

$$\sin(\tau + \tau_K) = \frac{n_u - \sin(\gamma) \cdot \sin(\delta)}{\cos(\gamma) \cdot \cos(\delta)} \tag{2.12}$$

If $\left|\frac{n_u - \sin(\gamma) \cdot \sin(\delta)}{\cos(\gamma) \cdot \cos(\delta)}\right| > 1$, there is no solution and with that there are no tracking angels, which can align the heliostat to the desired direction. Else there are two solutions for τ within $[-\pi, \pi]$. Since $\sin^{-1}(\phi)$ only returns values in $[-\frac{\pi}{2}, \frac{\pi}{2}]$, the first solution is always given by equation 2.13a. The second solution is given by equation 2.13b.

$$\tau_1 = \sin^{-1}\left(\frac{n_u - \sin(\gamma) \cdot \sin(\delta)}{\cos(\gamma) \cdot \cos(\delta)}\right) - \tau_K \tag{2.13a}$$

$$\tau_2 = \pi - \tau_1 - \tau_K \tag{2.13b}$$

From τ_1, τ_2, and the e- and n-component of the normal vector, the rotation angle of the first rotation axis, θ, can be determined. For that n_e and n_n are written as in equation 2.14, with $A = \sin(\gamma) \cdot \sin(\tau_1 + \tau_K) \cdot \cos(\delta) - \cos(\gamma) \cdot \sin(\delta)$, $B_1 = \cos(\tau_1 + \tau_K) \cdot \cos(\delta)$ and $B_2 = -B_1$.

$$\begin{pmatrix} n_e \\ n_n \end{pmatrix} = \begin{pmatrix} -B_{1,2} & A \\ A & B_{1,2} \end{pmatrix} \cdot \begin{pmatrix} \sin(\theta_{1,2} + \theta_K) \\ \cos(\theta_{1,2} + \theta_K) \end{pmatrix} \tag{2.14}$$

Inverting equation 2.14 yields equation 2.15, which can be solved by $\mathrm{atan2}(x, y)$, defined in equation A.2.

$$\begin{pmatrix} \sin(\theta_{1,2} + \theta_K) \\ \cos(\theta_{1,2} + \theta_K) \end{pmatrix} = \frac{1}{A^2 + B_{1,2}^2} \cdot \begin{pmatrix} -B_{1,2} & A \\ A & B_{1,2} \end{pmatrix} \cdot \begin{pmatrix} n_e \\ n_n \end{pmatrix} \tag{2.15}$$

From that, the solutions of θ are provided by equation 2.16.

$$\theta_{1,2} = \mathrm{atan2}\left(\frac{-B_{1,2} \cdot n_e + A \cdot n_n}{A^2 + B_{1,2}^2}, \frac{A \cdot n_e + B_{1,2} \cdot n_n}{A^2 + B_{1,2}^2}\right) - \theta_K \tag{2.16}$$

The determined angle pairs (θ_1, τ_1) and (θ_2, τ_2) require a validation check, since each heliostat defines a valid range for θ and τ, given by $[\theta_{min} - \theta_K, \theta_{max} - \theta_K]$ and $[\tau_{min} - \tau_K, \tau_{max} - \tau_K]$. If the first angle pair fits the valid range, it is used to calculate the according motor positions, otherwise the second pair is used, as long it is valid at all.

Up to here the normal vector $\vec{n}$ was considered as given. Actually it is an iterative process to calculate the normal vector and according tracking angles from a given sun vector $\vec{s}$ and aim point $\vec{a}$. Starting from the tracking angle pair $(\theta = 0, \tau = 0)$ the mirror center $\vec{m}$ is calculated following equation 2.9. In order for the heliostat to reflect a sunray to the desired target point,

the normal vector must form the bisector between the sun vector and the desired direction of the reflected sunray (see figure 2.3). The sun vector $\vec{s}$ is given as a unit vector and is equal for all heliostats in the field. Since the aim point $\vec{a}$ is defined from the ENU-COS' origin, the desired direction of the reflected sunray is given by $\vec{a} - \vec{m}$. From equation 2.17 the normal vector is calculated.

$$\vec{n} = \frac{\vec{s} + \frac{\vec{a}-\vec{m}}{\|\vec{a}-\vec{m}\|}}{\left\|\vec{s} + \frac{\vec{a}-\vec{m}}{\|\vec{a}-\vec{m}\|}\right\|} \tag{2.17}$$

Using the calculated normal vector, tracking angles θ and τ are updated and the calculation of the normal vector is repeated like this up to three times, what is sufficient for convergence in practice.

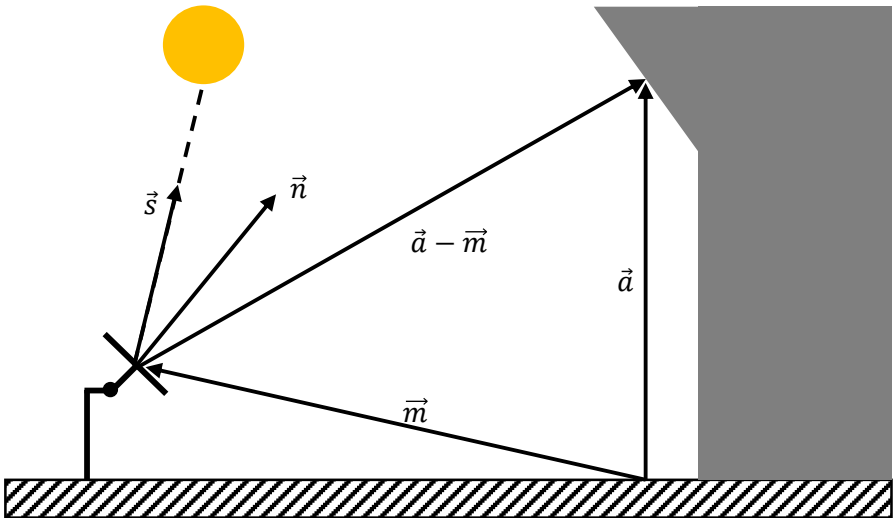


Figure 2.3: Sketch of the vectors used to calculate the normal vector of a heliostat

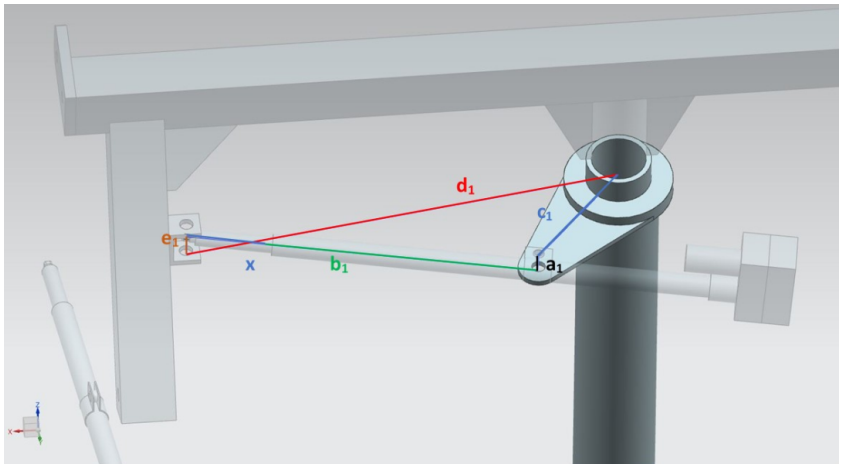
From the calculated tracking angles, the required motor positions of the actuators can be calculated. For that the geometry parameters of the linear actuators, as shown in figure 2.4, are required. The variable stroke length x determines the traverse angle φ of the corresponding axis, as shown in figure 2.5 as an example for a horizontal secondary axis. The traverse angle is a composition of the corresponding tracking angle $\Delta\varphi$, which is θ or τ, and a reference angle φ_0 given by $\varphi(x = 0)$. Using equation 2.18 the traverse angle can be determined by the stroke length, and vice versa the inverse function, given in equation 2.19, returns the stroke length for a given traverse angle.

$$\varphi(x) = \cos^{-1}\left(\frac{(b+x)^2 + a^2 - c^2 - d^2 - e^2}{-2 \cdot d \cdot c}\right) \tag{2.18}$$

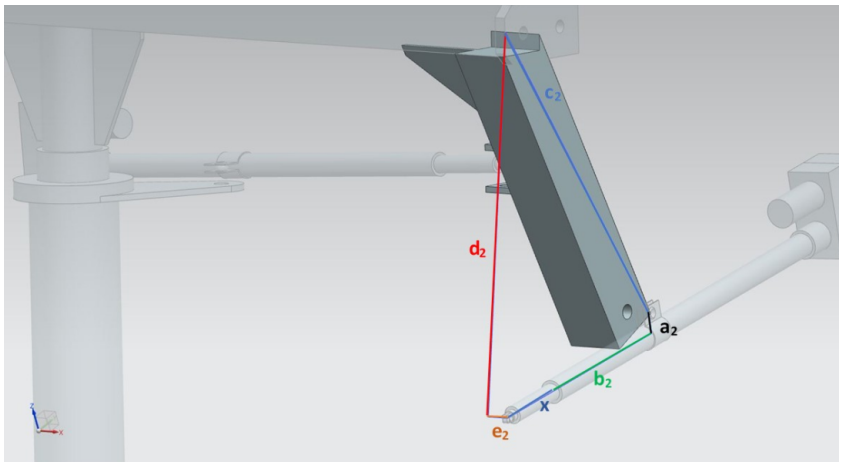
$$x(\varphi) = \sqrt{-2 \cdot d \cdot c \cdot \cos(\varphi) - a^2 + c^2 + d^2 + e^2} - b \tag{2.19}$$

Each actuator has a certain pulse ratio P, which defines the needed change in motor positions per meter. Finally, using the pulse ratio and stroke length, the motor position X of an actuator is determined as in equation 2.20.

$$X = \lfloor x \cdot P \rfloor \tag{2.20}$$

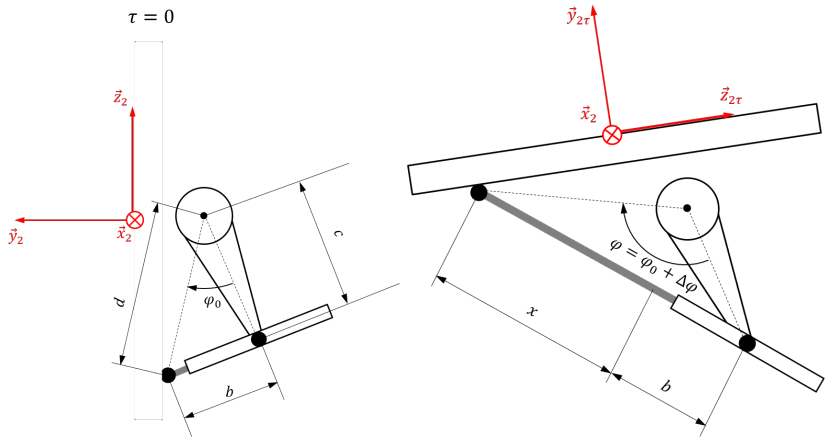


(a) Vertical primary rotation axis



(b) Horizontal secondary rotation axis

Figure 2.4: Kinematic heliostat geometry parameters of Mircon heliostat



(a) Motor fully retracted **(b)** Motor fully extended

Figure 2.5: Traverse range of a horizontal secondary axis

2.3 Controlling System for Heliostats

Inside the tower, there is the control station for the whole heliostat field. The server, that is located there, provides the absolute time, by the help of which the sun vector $\vec{s}$ is calculated using a Sun Position Algorithm (SPA), according to the Global Positioning System (GPS) coordinates of the heliostat field. For Jülich, the GPS coordinates are about (50.92 N, 6.36 E). The operator is able to access each heliostat using a Graphical User Interface (GUI). From the GUI each heliostat can be assigned to a focus point, and so the operation is controlled. Also the calibration process for certain heliostats can be started from the GUI. All the information about the heliostats, as position and geometry parameters, are stored in a Structured Query Language (SQL) based database on the server in the control station.

Each heliostat in the field holds a controller box that receives the motor positions that the two actuators should adopt. The motor positions for each heliostat are calculated on the control station server and are send to the regarding heliostat using a bus communication system. This means that every heliostat is connected to a node that has a single connection to the control station. There are several nodes distributed in the field, such that not every heliostat is connected to the same node.

In Jülich different heliostat types with different controllers and actuators from different manufacturers are installed. Many of the heliostats in Jülich are equipped with the Mircon system. The two linear actuators have a driving range from 0 (fully retracted) to 1995 (fully extended)

motor positions. The bulk of the heliostats are retrofitted with the Heliokon system, which has the advantage that the driving range of the actuators goes from 0 up to 78000 motor positions over the same maximal stroke length as the Mircon heliostats. For that the Heliokon heliostats can be set more accurately to the desired tracking angles (see table 2.1).

	Mircon	Heliokon
Axis1 Min. Counts [MP]	0	0
Axis1 Max. Counts [MP]	1995	78000
Axis2 Min. Counts [MP]	0	0
Axis2 Max. Counts [MP]	1995	65000
Pulse Ratio [MP/m]	3779.53	154166.67
Average Angular Pulse Ratio [mrad/MP]	0.98	0.024

Table 2.1: Comparison of heliostat drive system properties

An additional difference is, that the primary rotation axis of the Heliokon heliostats is horizontal ($\beta \approx 90°$), while the primary rotation axis of the Mircon heliostats are vertical ($\beta \approx 0°$). The basic orientation (motors fully retracted) of the two systems differ, since the normal vector of Mircon heliostat is orientated nearly horizontal in east-direction ($\alpha <= -90°$, $\theta_K = \tau_K \approx 0°$), while in a Heliokon system it is aligned more upwards with a tilt of around 54° from zenith towards east ($\alpha \approx 90°$, $\theta_K \approx 0°$, $\tau_K \approx 54°$). Figure 2.6 clarifies the resulting rotation axes.

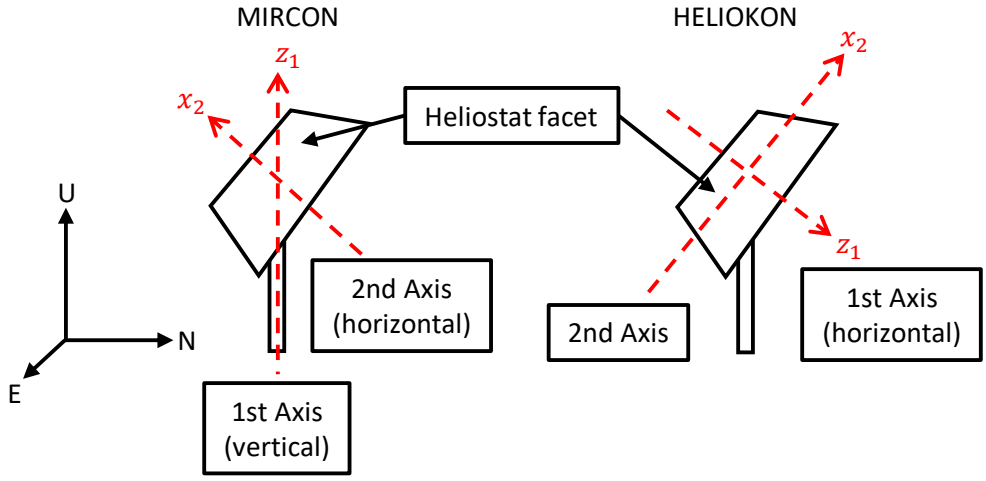


Figure 2.6: Comparison of rotation axes of Mircon and Heliokon heliostats

2.4 Causes of Incorrect Heliostat Alignment

To understand, why a calibration of heliostats is necessary to achieve a satisfactory tracking accuracy, an overview of causes of incorrect heliostat alignment is necessary. M. Hagenkamp has already elaborated this causes in his master thesis ([9]) and some of them are briefly summarized in the following. These are systematic errors that can be (partially) compensated by calibration. Statistical errors, such as wind loads, cannot be taken into account by a calibration system.

Angular inaccuracies When setting up and aligning the heliostat, it can happen that corresponding alignment angles are measured incorrectly. In addition, an unrecognized angular offset between the rotation axes (not orthogonal) can cause the heliostat to rotate about another unknown axis. Both can lead to large target deviations that increase proportionally to the distance from the target.

Position inaccuracies Due to the inaccurate determination of the heliostat position in the field or the axis distances, the position of the mirror center is calculated incorrectly. A deviation to east/west or up/down leads to an approximately equally large displacement on the target, while a deviation to north/south influences the calculation of the ideal angular alignment of the heliostat.

Motor resolution accuracy The accuracy of heliostat alignment is limited by the not infinite fine resolution of the motor drives. A high pulse ratio favors the alignment, since the displacement by one motor position results in a smaller angular change, and therefore the ideal alignment can be achieved better.

Wrong calculation of normal vector The desired normal vector of the heliostat's mirror surface is determined iteratively with a certain tolerance and also the sun position is calculated under uncertainties. From both follows a normal vector deviating from the ideal vector.

Deformation of the mirror facets and support structure Due to gravity and stressed mounting, which is favored by temperature fluctuations and wind loads, the mirror facets and also the underlying support structure deformate.

Miscount of the heliostat drives If the actual pulse ratio of the heliostat driving actuators is not constant and differ from the assumed value, wrong motor position are determined, leading to a wrong traverse angle and wrong heliostat alignment. The resulting error strongly correlate with the motor resolution accuracy.

Mirror tilt In order to focus the sunlight on an area smaller than the mirror surface, it is possible to tilt the mirror facets minimally (Canting). A minimal misalignment, may result in enlargement and displacement of the focal spot.

Aging Due to various signs of age, such as jamming of joints due to lack of lubrication and wear, or defective tooth edges, certain positions may no longer be approached.

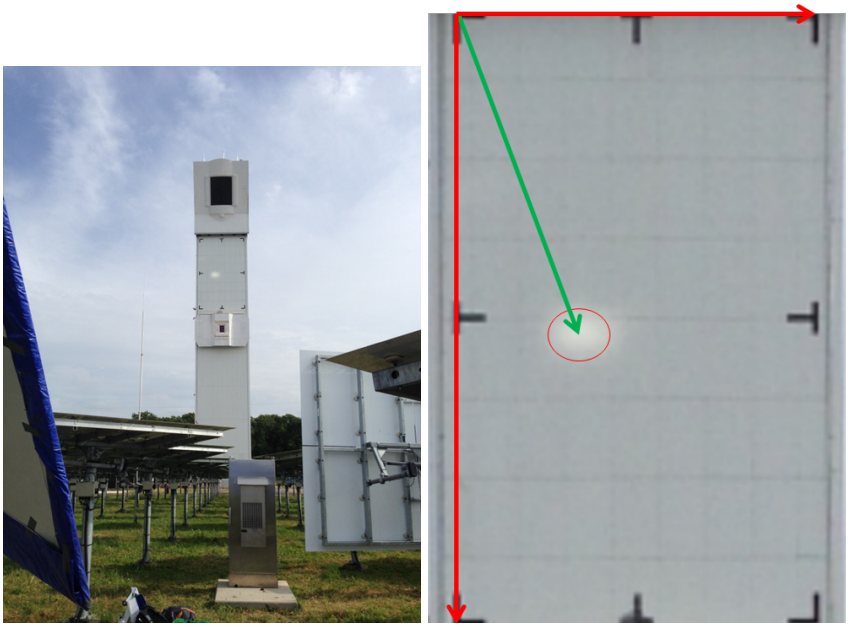
2.5 State of Calibration System

Before the calibration algorithm is described, the calibration system, which is required to collect data for the parameter optimization, is explained. As already mentioned, a system, referenced to the Camera-Target-Method is installed at the solar power plant in Jülich.

When talking about the Camera-Target-Method, several possible implementations can be meant. One way is to equip every single heliostat with an own camera and distribute several small targets on or around the tower [8]. Another way is to place a single camera in the heliostat field, observing the tower, holding one large extra target besides the receiver. In Jülich the extra target, called the calibration target is located directly underneath the main receiver (see figure 2.7a) and a simple 10-bit USB camera with a resolution of 720x480 pixel is available. The calibration target is bounded by eight markers (see figure 2.7b), whose coordinates are known. When a heliostat is aligned in a way such that the sun light is reflected on the calibration target, the focal spot can be seen on the target. Taking a picture with the camera, the exact position of the focal spot, also called the flux density focus, can be determined. For that the camera has to be calibrated according to the ratio between the flux density of the reflected focal spot and the diffuse ambient radiation, so that there is no overexposure or underexposure. Using the known camera position and the target markers position, the original taken image from the camera is rectified and mapped on the target plane. In the target plane there is drawn up a grid mesh, whose grid points are assigned a gray scale value averaged from the surrounding pixels. To get the flux density focus, first the gray scale value of each grid point is decreased by the background gray scale value. Subsequently the grid points positions, weighted by their gray scale value, are summed up and in the end divided by the sum of all gray scale values, which returns the flux density focus. Figure 2.8 shows the determined flux density focus for a given calibration image. For each

calibration image a new row in a database table, that stores information about all collected calibration images, is inserted, holding at least the following information:

- heliostat identifier
- calibration target identifier
- motor position of first rotation axis
- motor position of second rotation axis
- flux density focus (aim point) position in ENU-COS
- sun vector
- timestamp



(a) Original image **(b)** Rectified image

Figure 2.7: Calibration image taken at the solar tower in Jülich

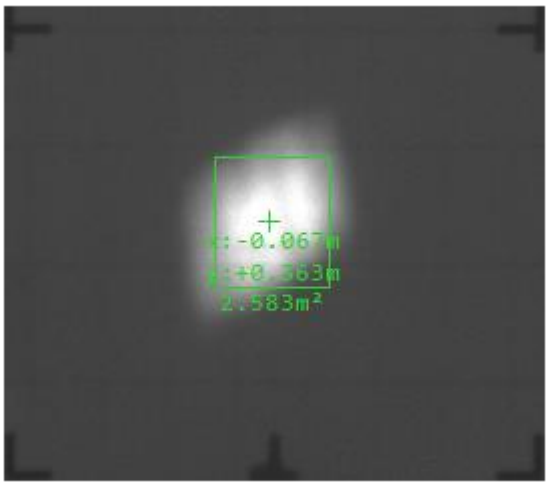


Figure 2.8: Determination of flux density focus

2.6 State of Calibration Algorithm

In order to improve the tracking accuracy of each heliostat and to compensate unknown error sources, selected parameters of the geometry model, described in section 2.2, are optimized. Let the vector $\mathbf{p}$ hold the parameters of the geometry model to be optimized. The goal of the parameter optimization is to minimize a - in this case nonlinear - target function F with respect to parameters $\mathbf{p}$. This optimization problem is defined in equation 2.21.

$$F\left(\mathbf{p}^{+}\right) = \min_{\mathbf{p}} F\left(\mathbf{p}\right) \tag{2.21}$$

Solving this problem, and find a set of parameters $\mathbf{p}^{+}$, that minimizes the target function globally over all possible values of the parameters, is very hard in general. For that reason, mostly a simpler problem of finding a local minimizer of F is solved, providing a minimum value of F inside a certain region. Starting from an initial guess for the parameters $\mathbf{p}_0$ the target function can be minimized by an iterative numerical method. Common methods are so-called descent methods. The main idea of this methods is to iteratively find a step $\mathbf{h}_k$ such that

$$F(\mathbf{p}_{k+1}) < F(\mathbf{p}_k) \text{ with } \mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{h}_k, \tag{2.22}$$

until the gradient of $F(\mathbf{p}_k)$ becomes smaller than a defined tolerance, which is close to zero, for any k. At that point $\mathbf{p}^*$ a local minimum of F is found [12]. The parameter optimization is the essential part of the calibration algorithm.

The following describes how the individual steps of the calibration algorithm proceed, which parameters of the geometry model are optimized and how the target function and the Levenberg-Marquardt (LM) algorithm, which is used to minimize the target function, are defined.

2.6.1 Calibration Cycle

Figure 2.9 shows the calibration cycle which is executed when the calibration is started by the operator. Only the error-free sequence is shown. The different states of the calibration cycle and possible sources of error are illustrated in the following:

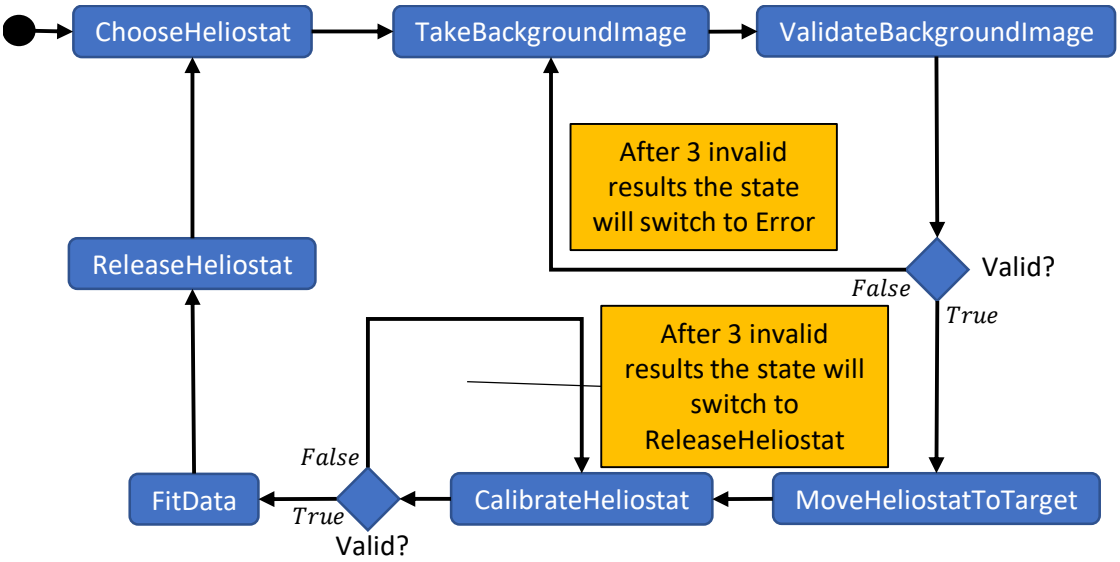


Figure 2.9: Calibration cycle in the software

ChooseHeliostat The calibration cycle starts with the choice of the heliostat to be calibrated. For that the operator has to select beforehand, which heliostats should be calibrated and put them in the so-called calibration pool. The choice is done automatically by an heuristic implemented algorithm that creates a calibration score (between 0.0 and 10.0) for each heliostat in the calibration pool, regarding the current number and age of calibration data and the last measured tracking accuracy. The heliostat with the lowest calibration score is the prefered choice. So if only one specific heliostat should be calibrated, only this one is put in the calibration pool.

TakeBackgroundImage and ValidateBackgroundImage When there is successfully found a heliostat to be calibrated, an image of the empty calibrated target (background image), as described in the section 2.5, is taken. This background image is checked for validity. First, there is checked if the Direct Normal Irradiance (DNI) is at least $200\frac{W}{m^2}$. In case of a low DNI an error message is send to the operator and the calibration is aborted. Secondly, the exposure is checked, and if the target plane is over- or underexposed, parameters of the camera, e.g. exposure time, is adapted and a new image is taken. After all it is tested, if the target is still occupied by another heliostat, by determining the standard deviation of the grey scale values. If this is the case, e.g. for a standard deviation of more than 10%, a certain time is waited and a new image is captured. While on failure of the first test, the calibration is aborted immediately, on failure of the second and third test a new image is captured until three invalid results are obtained before abortion.

MoveHeliostatToTarget The selected heliostat is decoupled from the actual power plant control and the focal spot is driven to the calibration target. Usually the origin of the calibration target is set as the aim point and the according motor positions are calculated from the currently valid geometry model.

CalibrateHeliostat As soon as the motor positions of both axes of the heliostat are within a certain tolerance of the nominal positions, the system automatically waits for a specified period of time and captures a calibration image of the calibration target with the same camera settings used for the background image. If the image is valid, which means that a focus spot of the heliostat is recognized, the calibration data are saved as described in section 2.5. A reason for invalidity could be that the heliostat focus spot does not lie (completely) on the target plane. In this case the operator is informed and the heliostat motor positions are adapted manually by the operator until the focus spot lies completely on the target and another image is taken. In other cases of invalidity (e.g. another heliostat is occupying the target), another image is captured after a certain time until there a three invalid results. When this occurs the heliostat is released and the next heliostat from the calibration pool starts the calibration process.

FitData and ReleaseHeliostat After capturing a valid calibration image and saving corresponding calibration data in the database, selected parameters of the geometry model are fitted to the calibration data, that are collected so far. The selection of fitting parameters depends on the number of valid calibration images of the heliostat. In this context valid means that a focus spot is recognized and the age of the image is not too high. In general, potential fitting parameters are the six angles θ_K, τ_K, α, β, γ and δ. The offset parameters θ_K and τ_K are fitted from the first calibration image, while with increasing number of calibration

images, the other parameters, first α and β then γ and finally δ, are added. In the current state, the scheme shown in table 2.2 is used for the selection of fitting parameters. As long γ and δ are not considered in the fitting, they are set to zero. The selected parameters are fitted towards the calibration data by minimizing a target function (see section 2.6.2) using a Levenberg-Marquardt-Algorithm implementation, described in section 2.6.3. The initial values for the fitting parameters come from the currently valid geometry model of the heliostat. From the optimization we get a new heliostat geometry model which is checked for validity, as explained in section 2.6.4. If the new model is valid, the fitted parameters are updated in the database, otherwise the old parameter are reused. After the fitting the calibrated heliostat is released and is either aligned regarding the assigned work aim point, using the fitted geometry model, or stalled in a defined stall position.

Number of valid calibration images	Fitting parameters
1 - 3	θ_K, τ_K
4 - 6	θ_K, τ_K, α, β
7 - 9	θ_K, τ_K, α, β, γ
> 9	$\begin{cases} \theta_K, \tau_K, \alpha, \beta, \gamma, \delta & \gamma > 1^\circ \\ \theta_K, \tau_K, \alpha, \beta, \gamma & else \end{cases}$

Table 2.2: Choice of fitting parameters in the current state of the calibration algorithm

2.6.2 Optimization Target Function

The target function F for the parameter fitting, is defined as a Mean Squared Error (MSE)

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (y_i - \tilde{y}_i)^2 \tag{2.23}$$

where in general, m is the number of observations, y_i the calculated output of a model with the input from the i-th observation and $\tilde{y}_i$ the measured output of the i-th observation. In the context of heliostat calibration, the observations are the calibration images with the observed global position of the focal spot for a given pair of motor positions and sun direction vector. Let the geometry of a heliostat be described by a geometry model $M(\mathbf{p})$ with fitting parameters $\mathbf{p}$. From the geometry model is calculated the global aim point $\vec{a}_{calc}$ of the heliostat on the target plane for given motor positions X_1, X_2 and sun vector $\vec{s}$. The calculated global aim point is then compared to the observed global position $\vec{a}_{meas}$ of the focal spot, by calculating the distance between them. Finally the target function with variables $\mathbf{p}$ is given by

$$F(\mathbf{p}) = \frac{1}{m} \sum_{i=1}^{m} \| \vec{a}_{calc}(M(\mathbf{p}), X_{1,i}, X_{2,i}, \vec{s}_i) - \vec{a}_{meas,i} \|^2. \tag{2.24}$$

2.6.3 Levenberg-Marquardt Algorithm

The LM algorithm is a damped iterative optimization method for non-linear least squares problems, combining the Gauss-Newton (GN) method with a regularization technique, to enforce descent function values [12].
In the context of non-linear least squares problems, the target function is formulated

$$F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m} f_i(\mathbf{x})^2 = \frac{1}{2}\mathbf{f}^T\mathbf{f} \tag{2.25}$$

with $\mathbf{f} \in \mathbb{R}^m$ being a vector holding all the m residual values f_i and $\mathbf{x} \in \mathbb{R}^n$ containing the n optimization parameters. Let $\mathbf{J}$ be the Jacobian matrix of $\mathbf{f}$ at a point $\mathbf{x}$. Using the chain rule, the gradient of F is given by

$$\mathbf{g} = \mathbf{J}^T\mathbf{f}. \tag{2.26}$$

In the GN method the residual $\mathbf{f}$ is linearized and inserted in F, leading to a quadratic model Q of F in the region of point $\mathbf{x}$ (see equations 2.27).

$$\mathbf{f}(\mathbf{x}+\mathbf{h}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}\mathbf{h} \tag{2.27a}$$

$$F(\mathbf{x}+\mathbf{h}) \approx Q(\mathbf{h}) \tag{2.27b}$$

To minimize F, iteratively a step $\mathbf{h}_{GN}$ minimizing Q, is determined. The derivative of Q is given by

$$Q'(\mathbf{h}) = \mathbf{g} + \mathbf{J}^T\mathbf{J}\mathbf{h} \tag{2.28}$$

and setting it equal to zero (sufficient condition for a minimum) yields the linear equation system

$$\left(\mathbf{J}^T\mathbf{J}\right)\mathbf{h}_{GN} = -\mathbf{g} \tag{2.29}$$

that only provides a unique solution for $\mathbf{h}_{GN}$ if $\mathbf{J}^T\mathbf{J}$ is positive definite (all eigenvalues are positive and non zero), which is also the necessary condition for $\mathbf{h}_{GN}$ being a minimizer of $Q(\mathbf{h})$, since $\mathbf{J}^T\mathbf{J}$ is the Hessian of the quadratic model. At this point the LM algorithm modifies the linear equation system towards the following

$$\left(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\right)\mathbf{h}_{LM} = -\mathbf{g}. \tag{2.30}$$

Using the damping parameter $\lambda > 0$ the system matrix is enforced to be positive definite, since the smallest eigenvalue takes at least the value of λ. For very small values of λ, $\mathbf{h}_{LM} \approx \mathbf{h}_{GN}$ and for large values $\mathbf{h}_{LM} \approx -\frac{1}{\lambda}\mathbf{g}$ [12].
Starting from an initial λ_0 the algorithm tries to find a step that minimizes the quadratic model and also ensures that the real target function is decreased. If latter is not given, the quadratic model does not fit the real function satisfactory in the region of the current point $\mathbf{x}$ and the damping parameter is increased, leading to a smaller stepsize. This is repeated until there is

found a decreasing step. The damping factor is then reduced again [12].
The current implementation of the LM algorithm used for the calibration is shown in algorithm A.1, where the damping parameter starts at 1 and is either increased or decreased by a constant factor of 10. The algorithm stops, when either the maximum number of target function evaluations - here 500 - is reached, the maximum absolute gradient entry is smaller or equal 10^{-12} or the relative change in the target function is smaller or equal 10^{-4}. Since the residual function is very complex the Jacobian $\mathbf{J}$ is approximated by a forward difference quotient

$$\mathbf{J}(\mathbf{x}) = \left(\frac{f_i(\mathbf{x} + \Delta \cdot \mathbf{e}_j) - f_i(\mathbf{x})}{\Delta} \right)_{i,j} \quad (i = 1, ..., m; j = 1, ..., n) \tag{2.31}$$

with a step size of $\Delta = 10^{-8}$. Any constant factor in the target function does not change the linear equation system in equation 2.30 since it cancels out. So the target function defined in section 2.6.2 is also applicable.

2.6.4 Validation of fitted geometry model

After the optimization, several checks are performed to validate the new model. First the target function is evaluated with the currently valid model and compared to the target function evaluated with the fitted model. If the target function is smaller using the fitted model, the fitted model provides a better fit of all collected calibration data points so far. Secondly the aim point with the given motor positions and sun vector from the very last collected calibration data point is calculated using both, the currently valid and the new fitted model, and compared to the actually observed aim point from the calibration image. It is desired that the new model provides a better result (smaller distance). Finally it is checked whether the fitted values of θ_K and τ_K are in a valid value range, given by $[-|\theta_{max} - \theta_{min}|, |\theta_{max} - \theta_{min}|]$ respectively $[-|\tau_{max} - \tau_{min}|, |\tau_{max} - \tau_{min}|]$. Only if all three tests are positive, the new model is valid and will be adopted.

3 Methods

The existing calibration system and calibration algorithm are already described in chapter 2. The task now is to find out how well the procedure works for the newly installed Heliokon heliostats and to work out improvement measures for the calibration algorithm. With the help of a simulation model, first a sensitivity analysis of various heliostat geometry parameters is carried out in order to be able to reconsider the selection of optimization parameters in the calibration. Subsequently, another simulation model is used to investigate whether erroneous parameters of a geometry model can be completely corrected by the existing calibration procedure, and other settings and optimization algorithms are tested.

3.1 Sensitivity Analysis of Geometry Model

To get a sense of how erroneous parameters affect the tracking accuracy of a heliostat, a sensitivity analysis is performed for different geometry parameters of the heliostat. The results of this sensitivity analysis will be used to derive adjustments to the calibration algorithm, especially the selection of the optimization parameters.

For this purpose, a geometry model with estimated parameters for the Heliokon heliostat BB.36, which is located approximately in the middle heliostat row centrally north in front of the solar tower, is generated. The parameters are listed in the Default Value column of table 3.1. Based on this geometry model, several perturbed models are generated, each with one different geometry parameter increased by 1 mrad (respectively 1 mm). The day of the summer solstice for the northern hemisphere, i.e. June 21, is selected as the time for the sensitivity analysis, since this day provides the most sunshine hours and thus the largest intervals for elevation and azimuth angles of the sun. In addition, times between 7 a.m. and 7 p.m. (CET = UTC+1) are considered. A simplified sun position algorithm is used to calculate the respective sun directions in normalized ENU coordinates for the location of the solar tower in Jülich. For each calibration, the heliostat is aligned so that the focal spot is on the calibration target. Therefore, the center (origin) of the calibration surface is always aimed at, which is located at $\vec{t} = (0.016, -3.235, 126.486)^T$ in the ENU-COS. So for each time considered, the required tracking angles $(\theta, \tau)_{set}$ and associated motor positions $(X_1, X_2)_{set}$ to reflect the sunlight onto the center of the calibration target are calculated

from a perturbed model. From the calculated motor positions the actual tracking angles $(\theta, \tau)_{is}$ of the unperturbed heliostat are determined and compared to the ideal tracking angles $(\theta, \tau)_{ideal}$, provided from the unperturbed heliostat model:

$$e_\phi = \phi_{is} - \phi_{ideal} \tag{3.1}$$

In addition the overall tracking error, defined as the angle between the actual and ideal normal vector is evaluated:

$$\vartheta = \cos^{-1}\left(\vec{n}_{ideal}^{T} \cdot \vec{n}_{is}\right) \tag{3.2}$$

Parameter	Default Value	Tolerance	Unit
θ_K	0.0	0.0873	rad
τ_K	0.94	0.262	rad
α	1.571	0.262	rad
β	1.571	0.262	rad
γ	0.0	0.0873	rad
δ	0.0	0.0873	rad
h_e	0.0	0.1	m
h_n	180.4	0.1	m
h_u	88.721	0.1	m
v_{12}	0.0	0.0	m
v_{2M}	0.172	0.03	m
a_1	0.0	0.0	m
b_1	0.07	0.02	m
c_1	0.335308	< 0.001	m
d_1	0.338095	< 0.001	m
e_1	0.0	0.0	m
a_2	0.0	0.0	m
b_2	0.07	0.02	m
c_2	0.340771	< 0.001	m
d_2	0.3191	< 0.001	m
e_2	0.0	0.0	m

Table 3.1: Geometry parameter values and tolerances of heliostat BB.36 from heliostat field in Jülich.
The defined tolerance values correspond to estimated values set by staff employed at the STJ based on empirical values.

The procedure described here yields the results shown in figure 3.1. On the upper two plots the errors e_θ and e_τ in the corresponding tracking angles are shown, while the plot on the

bottom represents the overall tracking error ϑ. Since obviously θ_K and b_1 only have an effect on θ, and τ_K and b_2 on τ, the cross-over errors are not considered in the figure, since they are zero. It is also clear that an error in θ_K or τ_K results in an opposed error with the same amount in θ or accordingly τ . But looking at the overall tracking errors induced by θ_K and τ_K, it is noticeable that the impact of an error in tracking angle θ on the overall tracking error ϑ varies over the day, while the impact of an error in τ seems to be constant. Interestingly, errors in γ and δ do only significantly effect the accuracy of tracking angle θ and the effects are very different over the day. In terms of geometry fitting in the calibration this means that errors in γ or δ can not be fully balanced by the variation of θ_K. Considering the errors caused by faulty α or β, they seem to have constant impacts on τ but verifying impact on θ over the day. So theoretically it would be possible to fully balance the error of either α or β with the variation of τ_K with respect to the error in τ but an error in θ would still remain. Up to this point, it can be concluded that none of the previous selected fitting parameters should be omitted from the calibration, although it should be kept in mind that in reality an error in γ and δ is almost zero. Accordingly, the relevance of these parameters in the optimization will be examined again in the later course of this thesis.

In addition to the previous fitting parameter, which are all angular values, also the kinematic parameters b_1 and b_2 of the driving geometry of each rotation axis are considered in the sensitivity analysis. The parameter b_1 belong to the first rotation axis and is therefore necessary to determine θ, while b_2 is needed to evaluate τ. Figure 3.2 presents the impact of faulty parameter b on angle θ or τ. It is clear to see, that an error in b does not result in a constant offset with respect to increasing motor position, which could be compensated by a variation of θ_K or τ_K. Instead the angle error increases quite strongly from a certain motor position. This effect is recognizable in figure 3.1 especially for the error in tracking angle τ. While the error in θ seems to be constant over the day with a faulty b_1, the error in θ increases continuously over the day with a faulty b_2. The reason why the error in θ does not seem to change is that the necessary motor positions for the elevation axis do only vary in a range of around 600 positions. In comparison, the motor positions for the azimuth axis change in range of 65000 positions. In view of these results, an additional optimization of the parameters b_1 and b_2 should be considered, whereby b_2 should be given a higher or equal priority, since an error in b_1 can be compensated by an adjustment of θ_K over a certain period of time, in which the range of the motor positions of the elevation axis to be adjusted hardly changes. The other parameters of the geometry drive geometry c and d are not considered in the sensitivity analysis since they have a significantly lower tolerance. Also the positional parameters h_e, h_u and h_u are not investigated, since a incorrect measure of the east- and up-coordinate of the position of the heliostat results in a equally large distance from the target point, which is significantly smaller than a target deviation caused by an incorrect angular parameter [9].

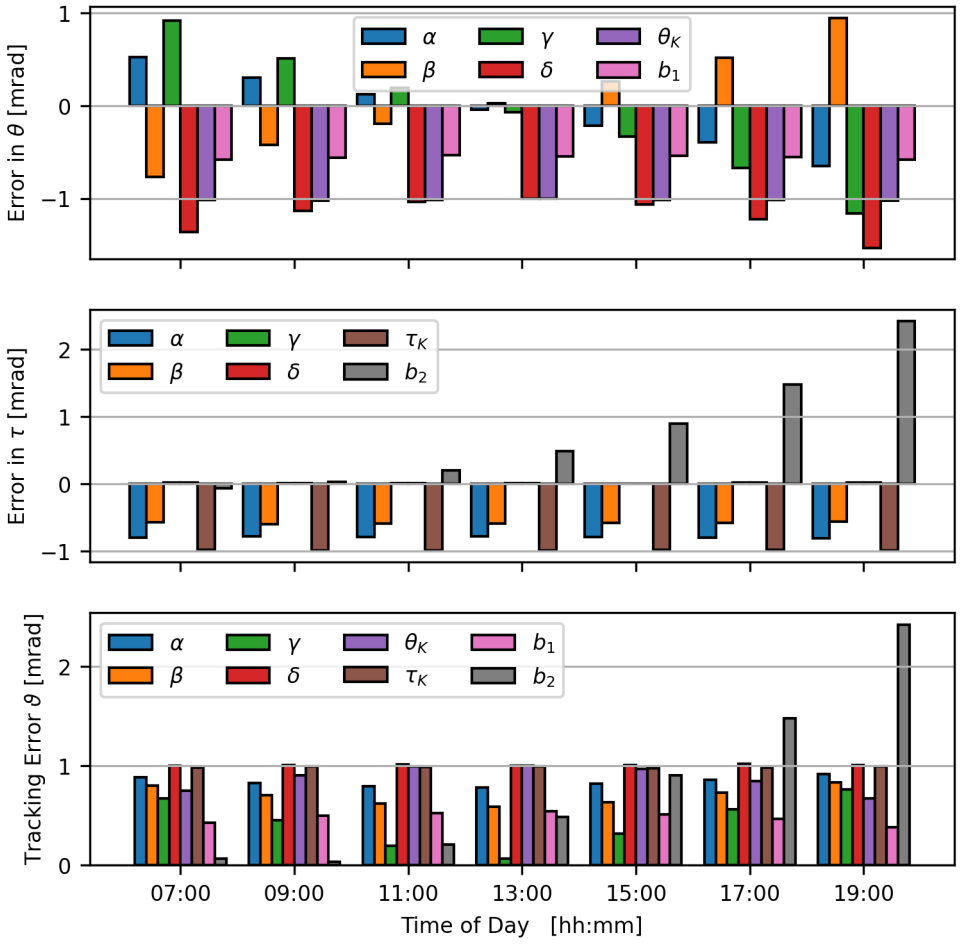


Figure 3.1: Tracking errors for Heliokon Heliostat BB.36 on 21.06.2021 with respect to a parameter error of 1 mrad (respectively 1 mm) of different geometry parameters

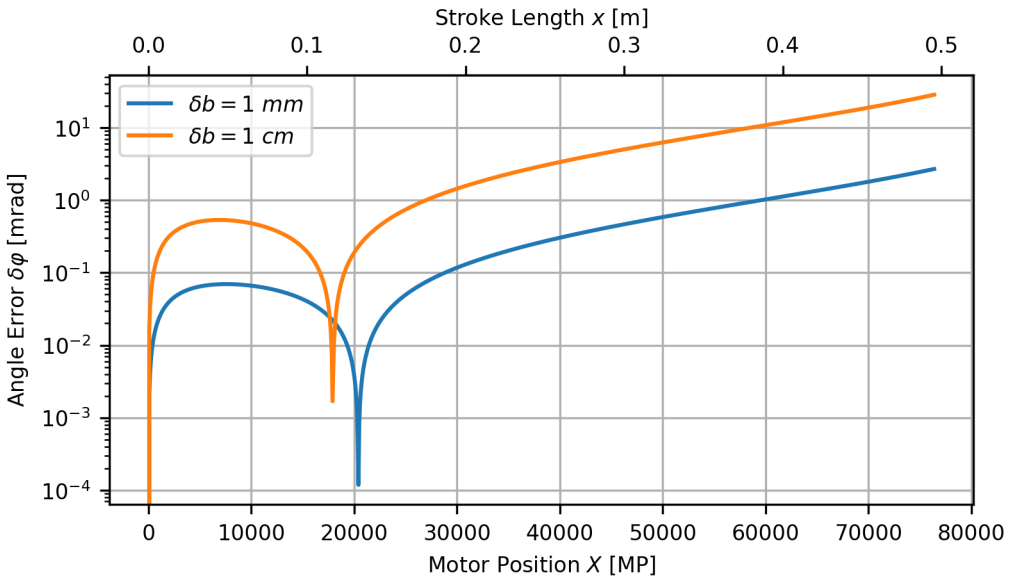


Figure 3.2: Resulting angle error on Heliokon heliostat with erroneous lever arm length for varying motor positions

3.2 Analysis of Calibration Algorithm

3.2.1 Simulated Calibration Cycle

In order to be able to estimate how well the calibration procedure works in reality, with various sources of error, a simulation environment is being developed in Python. The simulation environment provides all the kinematic functionality of a heliostat geometry, as defined in section 2.2, and the state-of-the-art fitting algorithm in the HeliOS software. In this environment, a heliostat geometry model is initialized, whose parameters are known, and is supposed to describe a real heliostat. Accordingly, it is assumed that the derived geometry model in section 2.2 can represent a real heliostat sufficiently well. At the same time, another geometry model is initialized, whose parameters are different from the model of the real heliostat, which should provide a first estimate for a fitted model. For the estimation values, the default values of Heliokon heliostat BB.36 from table 3.1 are chosen. Selected default values are perturbed with a random value within the specified tolerance and form the parameters of the real heliostat model. With the approach of using a simulation model with known parameters it is possible to investigate parameter convergence of a geometry model during the calibration cycle. In addition, a residual tracking error can be easily calculated after each calibration instead of having to take new calibration images, as is the case in reality.

For specified time points, the sun direction is determined for the location of the solar tower in Jülich and the real heliostat model calculates the required tracking angles and corresponding motor positions so that, for the given sun direction, the sunlight is reflected onto the center of the calibration target. From the calculated motor positions the actual tracking angles and finally the actual target point of the real heliostat are determined. The higher the resolution of the linear actuators, the more accurately the actual aim point corresponds to the center of the calibration target. Sun direction, motor positions and actual aim point are stored as calibration data for each time point considered. This procedure also is clarified by figure 3.3, whereby in this case the fitted model and the real model are identical. The fields highlighted in orange represent the stored calibration data which are divided into a fit and a test set. From that point the simulated calibration cycle, presented in figure 3.4, begins.

First there is calculated the initial tracking error ϑ_0 using the initial model and the test data, according to figure 3.3 with the aim point $\vec{a}$ being the stored hit point from the calibration data (calibration target origin). After that the fitted model is sequentially updated with an increasing number of fitting data points, starting from one fitting point up to the total number of fitting data points. After each iteration the tracking error is calculated again with the current model and test data.

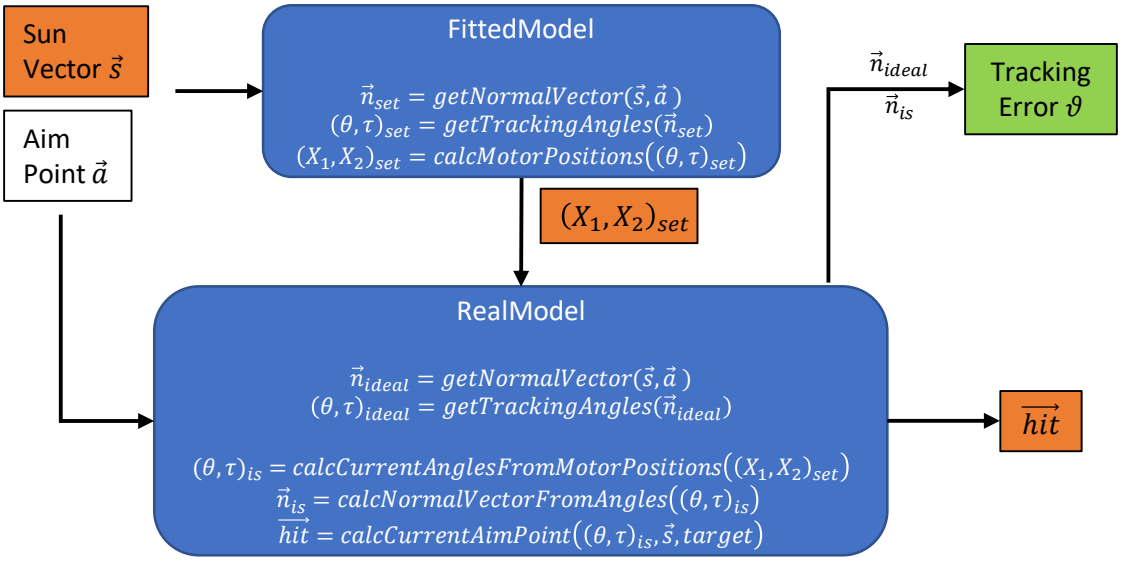


Figure 3.3: Creation of calibration data and calculation of tracking angle error from fitted geometry model and known real geometry model of a heliostat.
The values highlighted in orange are stored as calibration data, while the error in green is used for testing.

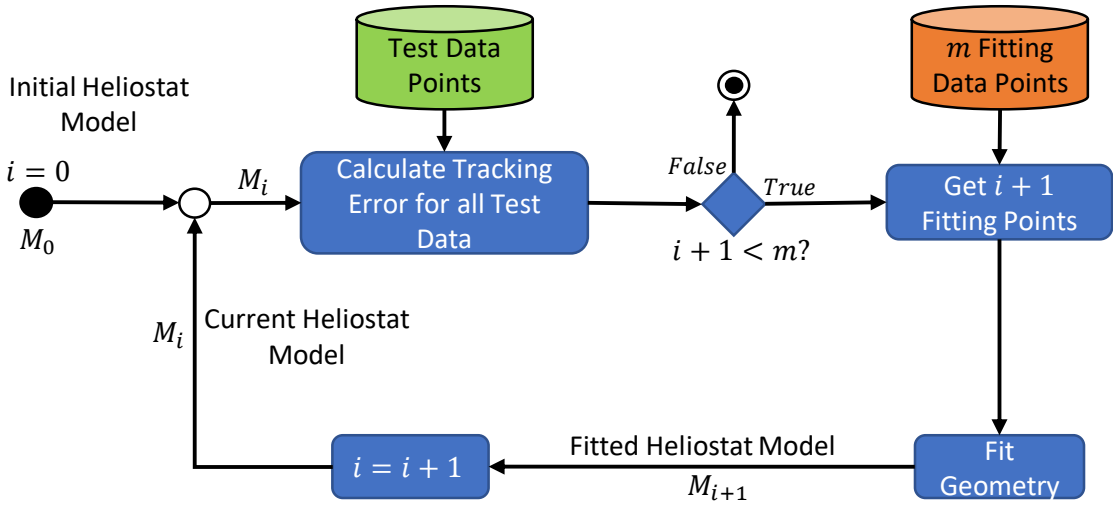


Figure 3.4: Flowchart of the simulated calibration routine

3.2.2 Analysis of Different Scenarios with Different Faulty Parameters

Best case scenario

The first case studied is a best case scenario where the initial estimated parameters of the fitting model already correspond to those of the real heliostat. In addition, the data used for the fitting are evenly distributed throughout the day of 21.06.2021 from 8 a.m. to 5 p.m. (CET) with a time interval of 20 minutes. The test data point are exactly between the fitting data points. The resulting sun position angles are shown in figure 3.5 with the sun angles defined as in figure 3.6.

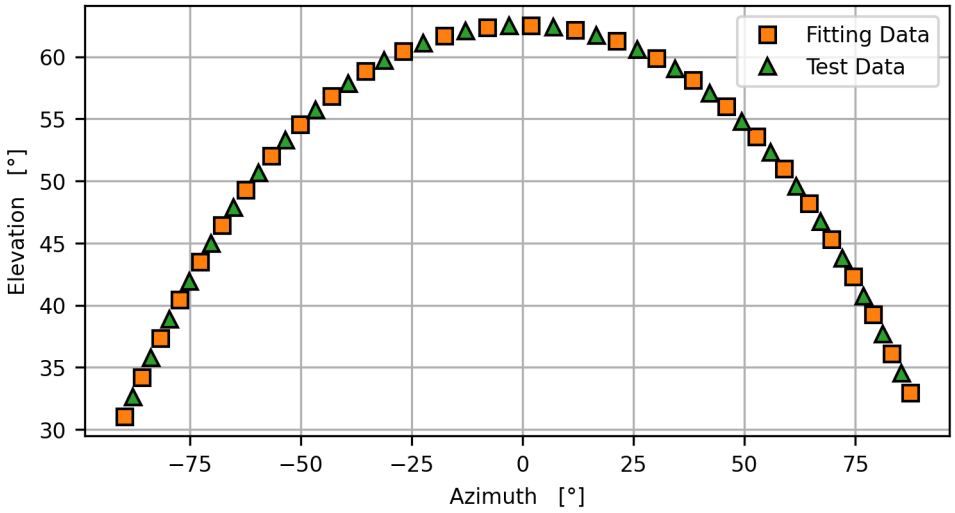


Figure 3.5: Choice of sun angles in the calibration data of best case scenario.
The evaluated sun angles are determined for the day 21.06.2021 in the time period from 8 a.m. to 5 p.m. (CET) for the location of the solar tower plant in Jülich, located at (50.91 N, 6.39 E).

In the calibration cycle, fitting data points are added in chronological order from early to late. According to the expectations, none of the parameters are varied, since they are already optimal. In the results of the tracking angle errors which are represented via a boxplot in figure 3.7, however, tracking errors of around 0.03 mrad occur, which can be attributed to the finite resolution of the linear actuators and corresponds to the order of magnitude of the angular impulse ratio from table 2.1. Based on this scenario, new error sources are gradually incorporated and the behavior of the calibration algorithm is investigated.

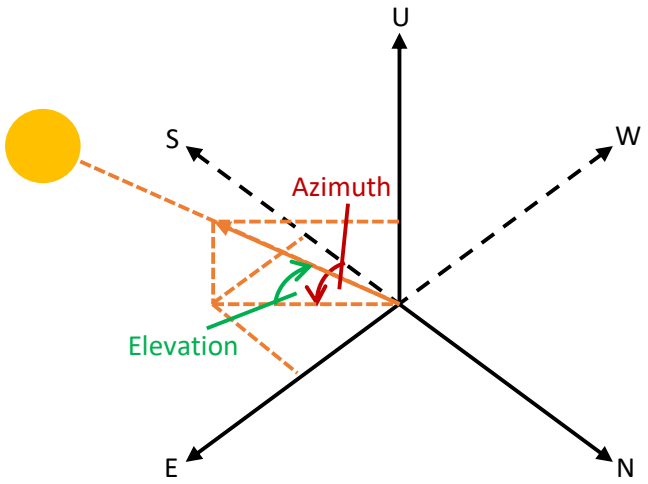


Figure 3.6: Definition of azimuth and elevation angle of the sun.
It is shown the definition of the azimuth and elevation angle of the sun in the ENU-COS. If the U-coordinate of the sun vector is positive then the elevation angle is positive else negative. The azimuth angle is positive if the E-coordinate is negative, else negative.

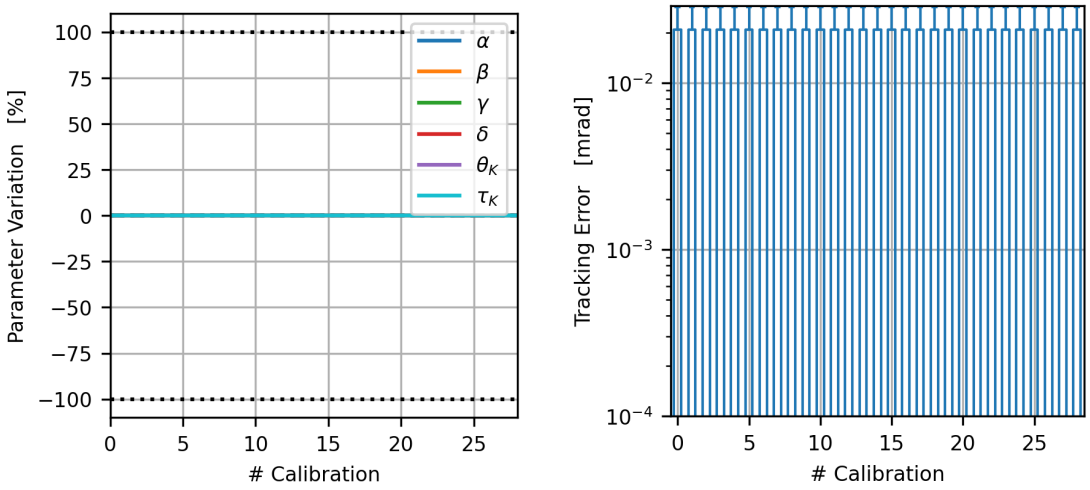


Figure 3.7: Calibrations results with initially exact parameters

Calibration with initially faulty θ_K and τ_K

With the same fitting and test data setup as in the section before, the calibration is started with disturbed values of θ_K and τ_K within the given tolerances in table 3.1. For that the NumPy random uniform number generator is used and to be able to reproduce the error values a random generator seed of 1 is set. Using this setup, values of $\theta_K = -0.01448\ rad$ and $\tau_K = 1.05536\ rad$ are created for the simulated real heliostat geometry model. Figure 3.8 displays the change of all optimization parameters during the calibration cycle.

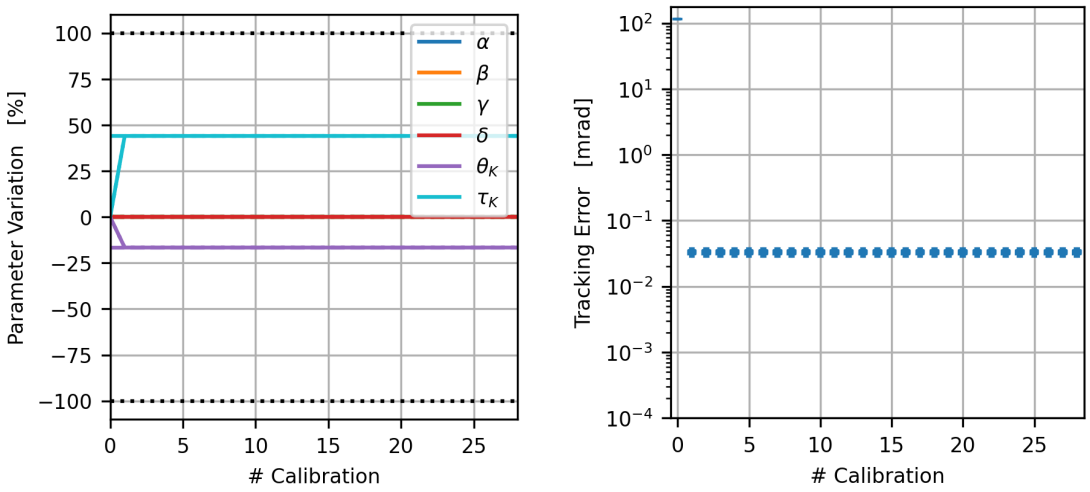


Figure 3.8: Calibrations results with initially faulty θ_K and τ_K

The difference to the initial estimated value, related to the tolerance range, is shown. It can be seen that the parameters θ_K and τ_K converge already after the first calibration and the real parameters are found with an accuracy of 10^{-11} mrad. This leaves again a tracking error of 0.03 mrad, after the initial tracking error had a value of more than 100 mrad (see figure 3.8).

Calibration with initially faulty θ_K, τ_K, α and β

Now the values for α and β are additionally disturbed with a random value within the tolerances, resulting in values of $\alpha = 1.3091\ rad$ and $\beta = 1.4673\ rad$, and the calibration is performed. It can be seen that it takes significantly more calibration points to find the exact parameters. Only with 27 calibration points the real parameters are achieved (see figure 3.9).

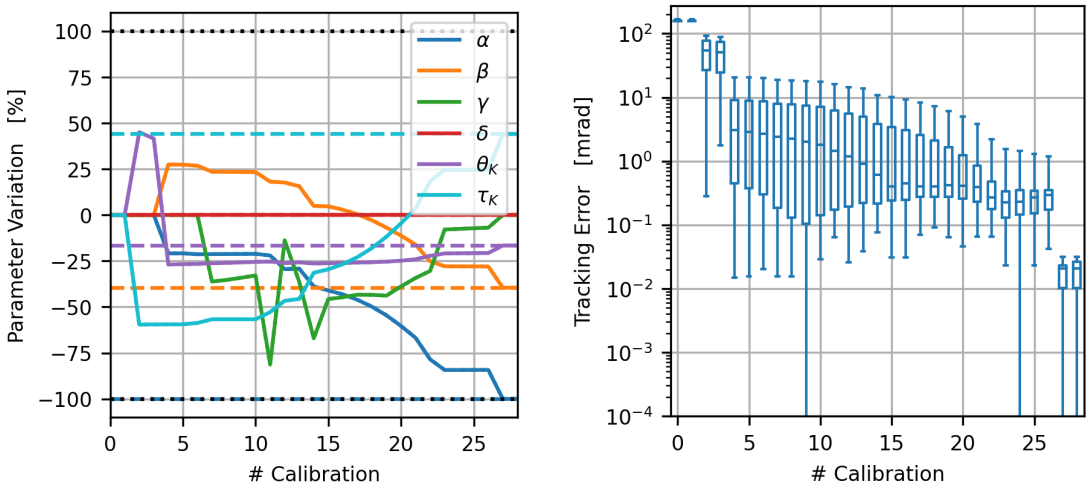


Figure 3.9: Calibration results with initially faulty θ_K, τ_K, α and β

Furthermore, no improvement of the model after the calibration with a single calibration point is recognizable. This has the background that with a singular system matrix of the equation system in equation 2.30, the LM algorithm is aborted and the parameters improved until then are discarded, so that the model is not changed. Mathematically, the system matrix has exclusively positive eigenvalues for each λ greater than 0, and therefore cannot be singular. Due to the limited accuracy of the used linear equation system solver of NumPy, very small eigenvalues are assumed to be zero. In this case λ takes a value of 10^{-12} after 12 iterations in the LM algorithm, which leads to a linear equation system recognized as singular. To counteract this effect, λ should be limited, for example to 10^{-8}, so that the smallest eigenvalue is always at least this value and is not declared as falsely singular.

Calibration with initially faulty θ_K, τ_K, α, β and γ

According to the previous investigations, γ is now also disturbed so that it takes a value of $\gamma = -0.01233\ rad$. Surprisingly, it turns out that in this case the optimization parameters converge much earlier against the exact parameters. Already after 11 calibrations the result is optimal (see figure 3.10).

Here even δ is varied in the meantime, although it is already correct from the beginning.

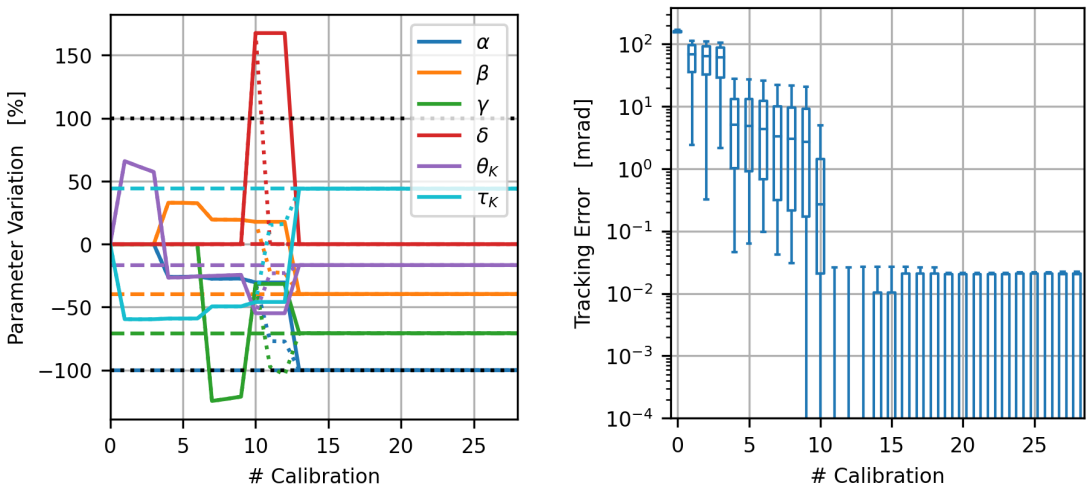


Figure 3.10: Calibration results with initially faulty θ_K, τ_K, α, β and γ

Calibration with completely initially incorrect optimization parameters

Finally, the behavior with completely disturbed parameters, summarized in table 3.2, is investigated.

Parameter	Value [rad]	Value [°]
θ_K	-0.1448	-8.296
τ_K	1.05536	60.468
α	1.3091	75.006
β	1.4673	84.07
γ	-0.01233	-0.706
δ	-0.01423	-0.815

Table 3.2: Disturbed parameter values of simulated real heliostat model in simulated calibration cycle

The exact parameters are not found for the given number of calibration points. The decisive factor is that delta is not optimized at all and is kept at zero, since gamma never exceeds the threshold value of 1 °, which is the decision criterion in the currently existing algorithm for whether delta is included in the optimization or not. Thus, after 28 calibrations, a maximum tracking error of approx. 0.3 mrad remains (see figure 3.11).

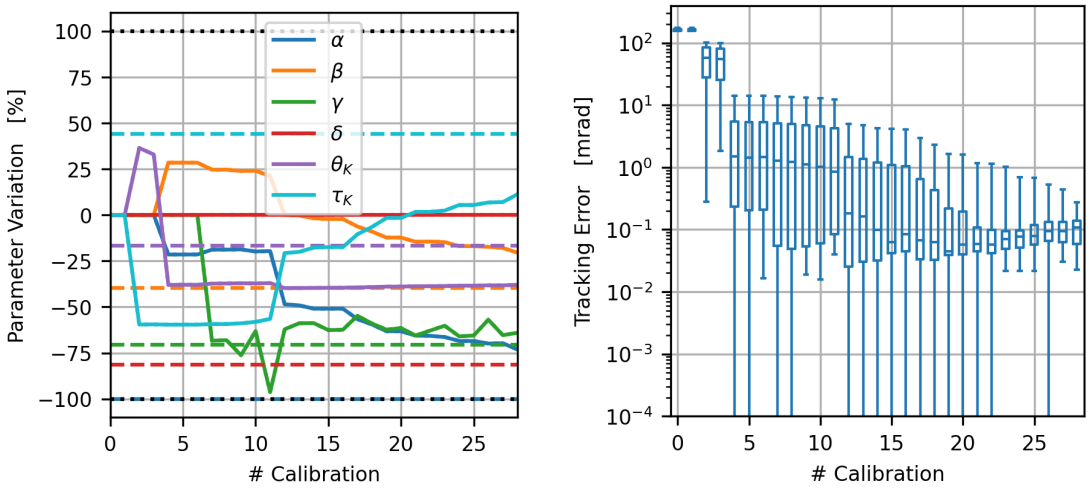


Figure 3.11: Calibration results with completely initially incorrect optimization parameters

3.3 Investigation of Improvement Measures

Since not all results from section 3.2 are satisfactory, because partly too many calibration images are needed to reach the parameters of the simulation model of the real heliostat, different approaches for faster convergence or better tracking accuracy are tested. In this procedure, the processes prior to optimization with the LM algorithm are modified first, before the LM algorithm itself is modified. Finally, the Python NumPy library is used to test a different optimization algorithm that, among other things, allows limits to be set for the optimization parameters.

3.3.1 Modification of General Optimization Settings

Reset Optimization Parameters to Default Values

In the current state of the algorithm, the LM algorithm always starts with the parameter values of the currently valid geometry model as the initial variable values. However, with a small number of calibration points, and thus a small number of selected optimization parameters, an attempt is made to compensate for errors in previously not optimized parameters. This may move the optimization parameters in a direction that does not correspond to the direction of their exact parameter value, as it happens for example in figure 3.9 with θ_K, τ_K and β. In general, it is difficult for local optimizers to find the global minimum, i.e. the exact parameters in this case, the further away the parameters are from their exact value, since there can

potentially be more local minima in between, at which the optimizer stops. One approach to counteract this is to reset the parameters to their default values before each new optimization. The approach is first tested on the case where θ_K, τ_K, α and β are in error. In fact, resetting the parameters before each calibration leads to a much faster convergence of the parameters. While without this adjustment 27 calibrations are necessary for convergence (cf. figure 3.9), with the adjustment only 12 are needed (see figure 3.12).

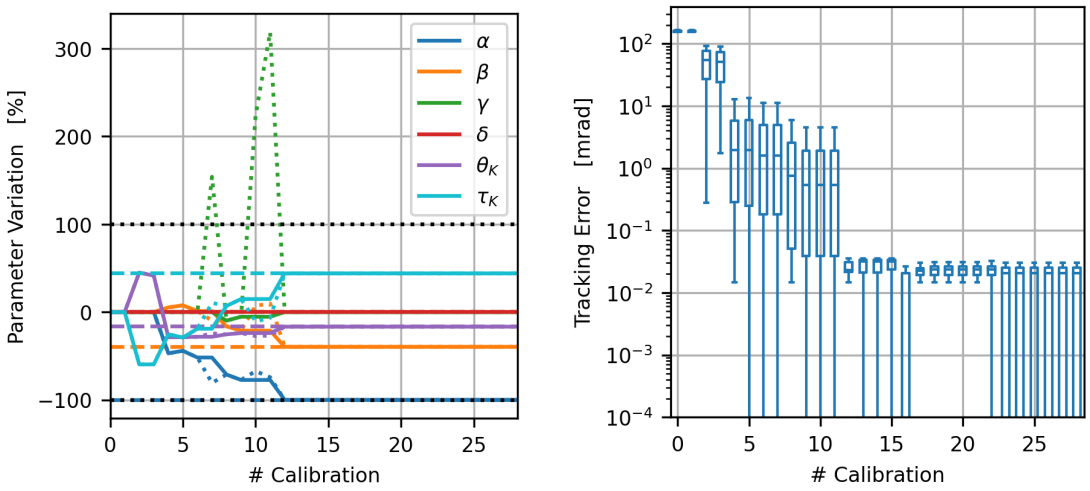


Figure 3.12: Calibration results with initially faulty θ_K, τ_K, α and β, and resetting parameters

It is also noticeable that the tracking angle errors are reduced faster with this method. If we repeat this test with the case where γ is also faulty, we can also observe a better convergence compared to the already very good one. Instead of 13 calibrations, 10 are sufficient (see figure A.1). In the case of completely incorrect parameters, the tracking error can be reduced a bit more partially with this setting, but the exact parameters are not found, because δ is still not considered in the optimization (see figure A.2). In order to get the best results in this case, the selection of optimization parameters should be additionally adjusted, which will be investigated in the next section. As a conclusion, however, it can be said that resetting the parameters - at least in this example - has a significant positive influence on the convergence speed against the exact parameters. Thus, this approach can definitely be considered as an improvement measure.

Adjustment of the Optimization Parameter Selection

In the previous investigations it was already stated that it is difficult, or even impossible, to find the exact parameters in the optimization if the parameters are completely disturbed, since

δ may not be included in the optimization and is kept at zero. Therefore, it is appropriate to investigate at this point how the calibration results behave when δ is always optimized independently of the value of γ from a certain number of calibration points. Currently γ is included in the optimization from the 7th available calibration point, δ from the 10th available calibration point at the earliest, according to table 2.2. In practice, it is assumed that γ and δ always have a value close to 0, so that a later integration of these parameters can have a positive effect on the optimization of the other parameters. In general, fitting parameters can be solved much better if the number of fitting data is much larger than the number of parameters, which also speaks for a later integration of not so relevant parameters. In a first test it is checked whether the faulty parameters θ_K, τ_K, α and β are better corrected if the non-faulty parameters γ and δ are not optimized at all, and how many calibrations it takes to find the exact parameters. With the values from table 3.2 and without resetting the parameters before each calibration the effect of only optimizing faulty parameters is very low, since only two calibrations less, namely 25, are needed (see figure A.3). However, when resetting the parameters, the effect is huge and only 7 calibrations are necessary, compared to 13 (see figure 3.13).

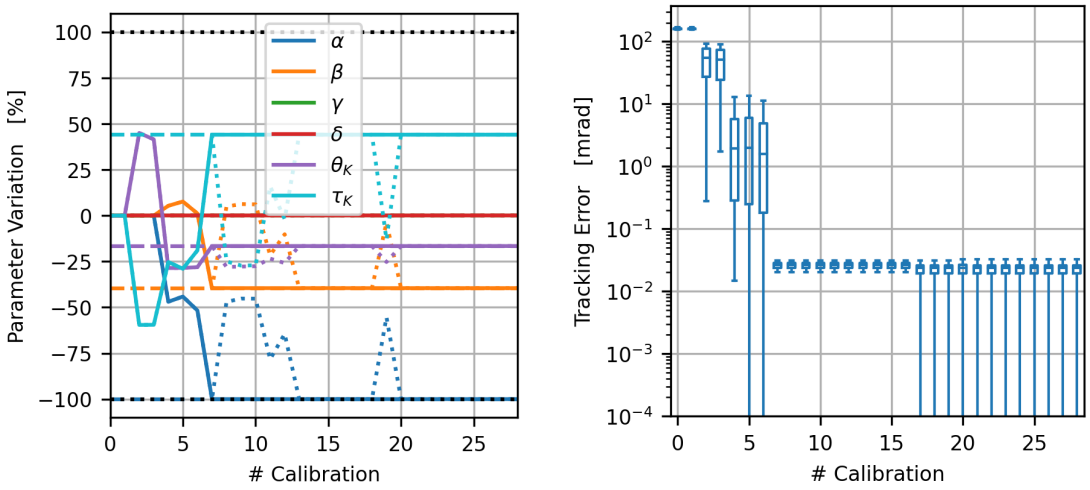


Figure 3.13: Calibration results with only optimizing initially faulty parameters θ_K, τ_K, α and β, and resetting parameters

According to the overall results, a configuration for the selection of optimization parameters is chosen as in table 3.3, such that convergence of the first four fitting parameters after 7 calibrations is preserved, if only them are faulty, and also convergence with additionally faulty γ after 10 calibrations is kept.

In order to possibly achieve convergence even with faulty δ, this parameter is included in the optimization from the 13th calibration point onwards, regardless of the value of γ. Using this

Number of valid calibration images	Fitting parameters
1 - 3	θ_K, τ_K
4 - 9	θ_K, τ_K, α, β
10 - 12	θ_K, τ_K, α, β, γ
> 12	θ_K, τ_K, α, β, γ, δ

Table 3.3: Alternative selection of fitting parameters in the calibration algorithm

configuration in combination with the reset of optimization parameters before each calibration, a convergence of totally initial faulty parameters towards their exact values is achieved after 14 calibrations, as one can see in figure 3.14.

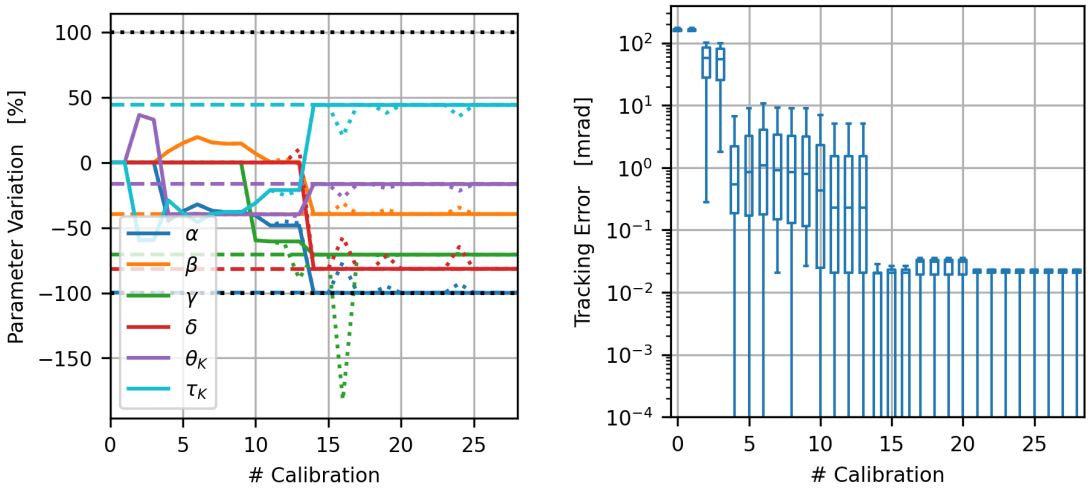


Figure 3.14: Calibration results with completely initially incorrect optimization parameters, resetting parameters and alternative optimization parameter selection configuration

Investigating again the case with only θ_K, τ_K, α and β being faulty, and the configuration from table 3.3, it can be seen that after reaching the exact parameters, other parameters are found in subsequent calibrations (see dotted lines in figure 3.15), but these are not adopted because the model with the exact parameters has a lower value for the target function (see section 2.6.4), such that the model with exact parameter is always kept as the valid model used to track the heliostat.

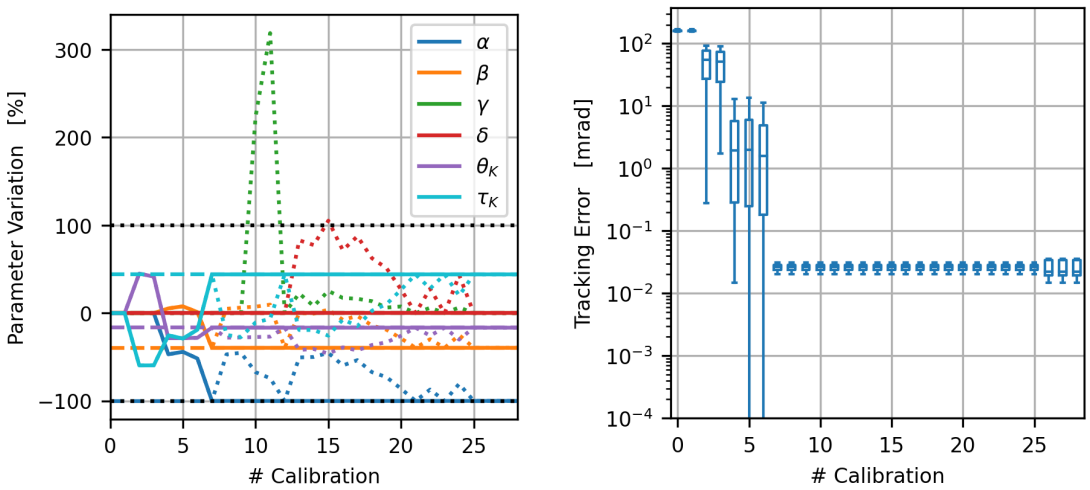


Figure 3.15: Calibration results with initially faulty parameters θ_K, τ_K, α and β, resetting parameters and alternative optimization parameter selection configuration

Results with other parameter values

In order to make a more general statement about the investigated measures taken so far, the case with erroneous θ_K, τ_K, α and β is repeated with other parameter values for the simulated real heliostat model. For this purpose, the seed of the random number generator is varied between 0 and 4, so that five different cases, including the case considered so far, can be compared. The random generated values to the corresponding seeds can be found in table A.1, whereby the values of γ and δ are set to 0. From table 3.4 can be seen the number of calibrations needed to find the exact values of the faulty parameters for different seeds and setups. A '-' indicates that the parameters are not found within the considered number of calibrations. In general, it can be said that adjusting the parameter selection together with resetting the optimization parameters tends to lead to a visible improvement in most cases (here in 4/5 cases), whereas adjusting the parameter selection alone leads to no improvement or even a deterioration in most cases (here also in 4/5 cases). For that reason, resetting optimization parameter to default value before each calibration should be considered in the real calibration process.

3.3.2 Modification of Levenberg-Marquardt Algorithm

Up to this point, only settings that affect the processes before the actual optimization algorithm is called have been considered. The LM algorithm also offers great scope for change, which

Seed	# Calibrations
0	7
1	27
2	4
3	15
4	21

(a) Current settings

Seed	# Calibrations
0	7
1	-
2	4
3	-
4	10

(b) Adjusted parameter selection

Seed	# Calibrations
0	6
1	7
2	10
3	4
4	10

(c) Adjusted parameter selection + resetting

Table 3.4: Number of necessary calibrations until exact parameters of θ_K, τ_K, α and β are found for different settings

will be examined in the following. In the literature is mainly dealt with the selection of the initial damping parameter and its update strategy [12, 20], which will be limited to in this work.

Update Strategy of the Damping Parameter

As described in section 2.6.3, the damping parameter λ is decreased, when a calculated step is successful, i.e. reduces the target function, and else increased. In the current state λ is equally decreased or increased by a factor of 10, which is also the MathWorks® approach in their implementation of the LM algorithm [3]. Another widely used strategy is to decrease λ by p_1 and increase it by p_2 with $p_1 \neq p_2$ and $p_1 < p_2$, such that the values of λ cannot oscillate [12]. Common value pairs (p_1, p_2) are (2, 3) or (1.5, 5) [20]. Furthermore Nielsen demonstrated the following strategy including the so-called gain ratio

$$\rho = \frac{F(\mathbf{x}) - F(\mathbf{x}+\mathbf{h})}{Q(\mathbf{0}) - Q(\mathbf{h})}, \tag{3.3}$$

i.e. the ratio between the actual and predicted decrease of the target function value:

$$\begin{aligned} &\textbf{if } \rho > 0: \\ &\quad \lambda = \lambda \cdot max\left(\frac{1}{3}, 1-(2\rho-1)^3\right); \quad \nu = 2 \\ &\textbf{else}: \\ &\quad \lambda = \lambda \cdot \nu; \quad \nu = 2 \cdot \nu \end{aligned} \tag{3.4}$$

The factor ν is initialized to $\nu = 2$, and the statement $\rho > 0$ means that the actual target is reduced since the denominator in 3.3 is always positive by definition [12]. Using the parameter values from table 3.2 for the real heliostat and the corresponding default values from 3.1 as initial guess, the calibration cycle is performed with resetting optimization parameters and

adjusted selection of optimization parameters (see section 3.3.1). In each calibration the LM algorithm is executed four times with the four different update strategies of the damping factor and the resulting values of the target function are compared. In figure 3.16 it can be seen that the choice of update strategy is unheralded for a small number of calibration dates and in this case only from the 13th calibration onwards, when all parameters are also included in the optimization, there is a significant difference in the results of the remaining target function value.

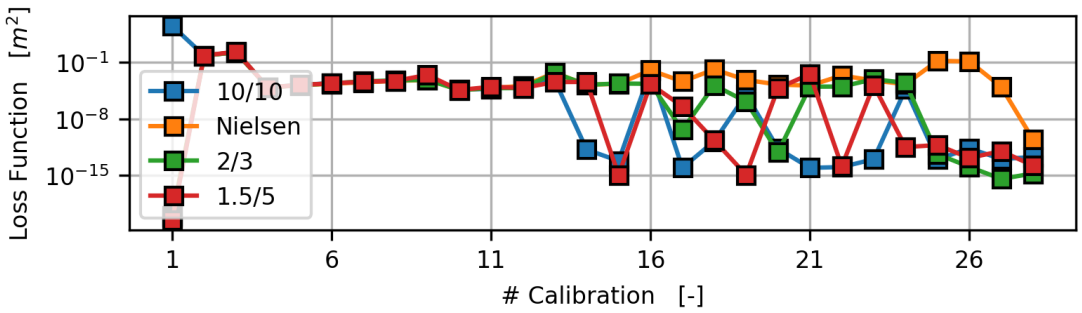


Figure 3.16: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied update strategy of the damping parameter λ for real geometry parameter values generated with random generator seed 1

In this case, we can say that the strategy used so far (10/10) is superior to the other methods, since with this method the minimum of the objective function, which is 0, is most stably approximated, which is equivalent to the convergence of the optimization parameters against their exact values. Repeating this investigation with other exact parameter values from seeds 0 and 2 (see table A.1), the same statements can be made as in the case considered here, so that the update strategy used so far can be described as suitable for the optimization problem at hand (see figures A.4 and A.5).

Initial value of the damping parameter

In the current state of the algorithm, the LM algorithm is constantly initialized with a damping parameter value of 1. To check the sensitivity of this parameter, it is first varied between 100 and 0.01, and as above, an optimization is performed for each case during the calibration cycle. Again, in figure 3.17 it appears that a change in this parameter makes no significant difference with few calibration data.

For an increasing number of calibration data it can be seen that with increasing initial values, the LM algorithm is less successful. For small initial values, there is a tendency to show an

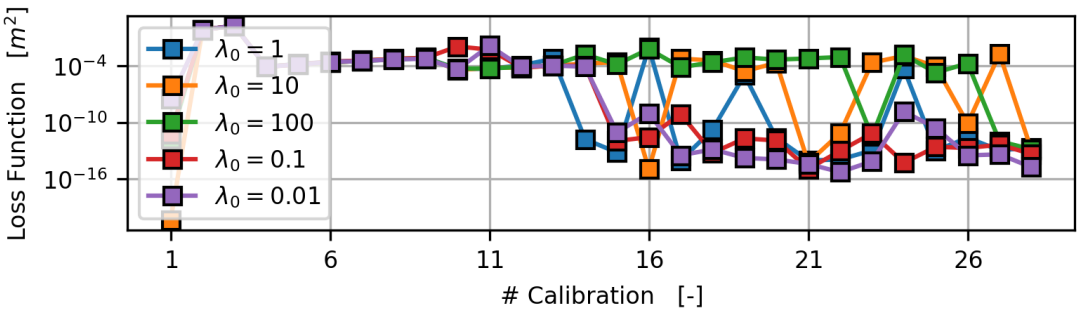


Figure 3.17: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied initial value of the damping parameter λ for real geometry parameter values generated with random generator seed 1

improvement, although this is not really significant, since the resulting value of the objective function is in the range between 10^{-10} and 10^{-17}. The same behaviour appears for different parameter values (see figures A.6 and A.7).
An alternative way, submitted by Madsen [12], is to set the initial value of λ in relation to the the largest diagonal value of the matrix $\mathbf{A}_0 = \mathbf{J}(\mathbf{x}_0)^T\mathbf{J}(\mathbf{x}_0)$ by providing a value ξ such that

$$\lambda_0 = \xi \cdot \max(diag(\mathbf{A}_0)) \tag{3.5}$$

Common values for ξ are $\xi = 10^{-6}$, $\xi = 10^{-3}$ or even $\xi = 1$, where the more it is believed that $\mathbf{x}_0$ is a good approximation to the optimal solution, the smaller the value. As shown in figures 3.18, A.8 and A.9, $\xi = 10^{-6}$ seems to be a good choice, since it competes with the reference case where $\lambda_0 = 1$ the best and provides a dynamic setting according to the considered case.

3.3.3 Investigation of SciPy Least-Sqaures Problem Solver

So far, a basic implementation of the Levenberg-Marquardt algorithm has been used and studied, in which the step size is indirectly controlled (damped) by adjusting the damping parameter λ, but not explicitly limited. The SciPy optimization library provides several optimization methods for least squares fitting problems, including a very robust version of the LM algorithm based on the reflections of Moré [13], implemented in the MINPACK library [14]. In simplified and rough terms, the basic difference is that the damping parameter λ is varied, such that the determined step $\mathbf{h}_{LM}$ from equation system 2.30 is inside a trust-region Ω, i.e. $\|\mathbf{h}_{LM}\| <= \Omega$, where either $\lambda = 0$ and $\|\mathbf{h}_{LM}\| <= \Omega$, or $\lambda > 0$ and $\|\mathbf{h}_{LM}\| = \Omega$. According to the gain ratio ρ the trust-region radius Ω is either decreased or increased. Many smart tricks are used to achieve high robustness, which are not described in this

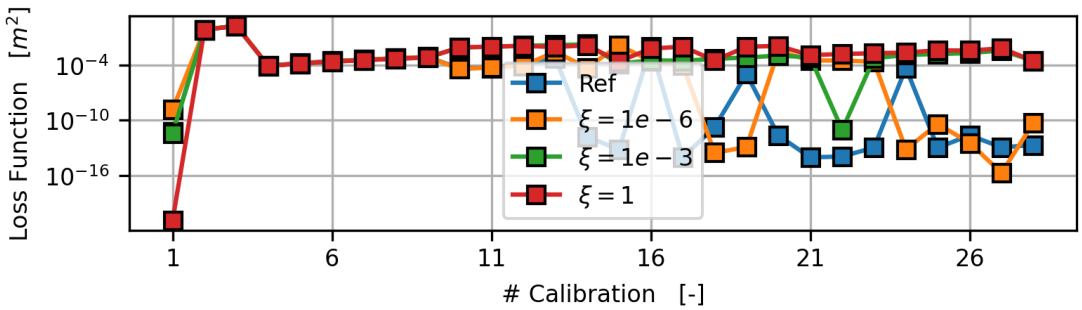


Figure 3.18: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied factor ξ relating the initial damping parameter λ to the initial Jacobian values of the target function for real geometry parameter values generated with random generator seed 1

thesis. The other provided optimization methods provided by SciPy are the Trust Region Reflective (TRF) algorithm, formulated as in [7], and a modified version of Powell's Dog Leg (DL) method, which works with a combination of Gauss-Newton step $\mathbf{h}_{GN}$ (see equation 2.29) and the steepest descent direction $-\mathbf{g}$, also inside an adaptive trust-region [12], which is a hyperrectangle instead of a hypersphere in the modified version [22]. Both methods are able to consider bounds for the optimization variables, while the LM method is not.

To check the performance of the mentioned SciPy optimization methods and compare it to the LM implementation that is used so far, all six fitting parameters are optimized in the calibration cycle from the first calibration point using each of the mentioned SciPy solver. For the TRF and DL method also the bounded optimization is considered, using the tolerances of the parameters defined in table 3.1. Figure 3.19 clearly shows that all SciPy methods, except the bounded TRF, outperform the LM algorithm regarding the number of calibration data needed to find the optimal solution (exact parameters) with a loss function value of around 10^{-11}.

This result also appears, when we change the exact parameters using seed 0 and 2 for the random generator. The corresponding plots are added in the appendix (Fig.A.10, Fig.A.11). With this insight, it can be deduced that when using one of the SciPy trust region optimization methods, more parameters can be included in the optimization much earlier, or even all six parameters can be optimized from the beginning. Using a bounded optimizer, e.g. the DL with hyperrectangular trust regions, which easily can be adapted to defined parameter tolerance bounds, can help to achieve valid parameter values.

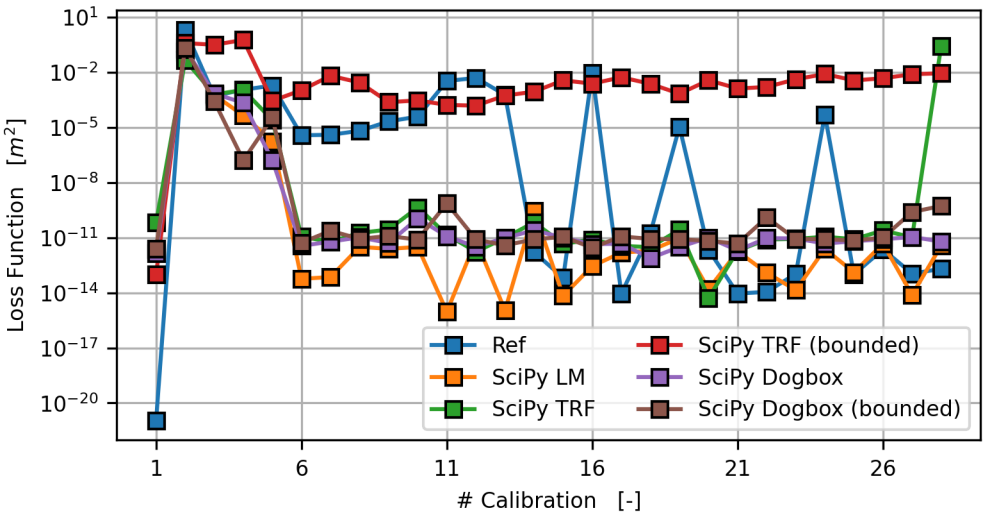


Figure 3.19: Comparison of the different SciPy least-squares problem solvers with the current Levenberg-Marquardt algorithm

4 Results at the Solar Tower in Jülich

In the previous part of the thesis, only simulated calibration data are considered with known parameters of the real heliostat, which are not exactly known in reality. At the solar tower power plant in Jülich there has already been collected some calibration data of the recently commissioned Heliokon heliostats. Using the results from the previous chapter, different geometry fitting strategies are derived and compared using the collected calibration data. As in the simulated calibration cycle, the calibration data are inserted one after the other in ascending chronological order and after each insertion a geometry fitting is absolved. After each fitting, the resulting geometry model is tested, looking at all the collected calibration points, those that have already been included in the calibration and those that will be integrated later. In the test, for each calibration point, the motor positions for the given sun direction and observed focal spot on the calibration target are calculated from the currently valid geometry model and compared to the motor positions that were set when the calibration image was acquired. A comparison of the calculated and ideal normal vector is not possible here because the exact heliostat parameters are not known. The deviation of the calculated motor positions provides an indication of the tracking accuracy of the heliostat. Using the average angular pulse ratio of $0.024\ \frac{mrad}{MP}$ an approximate tracking error can be inferred.

Here the collected calibration data of the heliostat AQ.64 are used to test different approaches. This heliostat counts 21 calibration images, collected between the 31.03. and 14.04.2021 as shown in figure A.12. The representation of the calibration data shows that the input data, as sun position and resulting motor positions vary sufficiently, while the target position is always the origin of the target plane. The heliostat is located approximately 123 meters east and 110 meters north from the solar tower and the geometry model of this heliostat is initialized with the values from table A.2.

First, four strategies are defined, which test the adjusted parameter selection derived in section 3.3.1 (ALT1-ALT4 in table 4.1) and compare them to the State-of-the-Art (SOTA) algorithm. The four cases differ in the choice of optimization algorithm, deciding between the LM algorithm with adaptive initial damping parameter and the Dogbox algorithm provided by SciPy. Furthermore, it is varied whether the fitting parameters are reset to their default value before each new optimization.

	Integrate parameter in calibration from calibration image no.										
	θ_K	τ_K	α	β	γ	δ	b_1	b_2	Reset?	Optimizer	Optimizer Settings
SOTA	1	1	4	4	7	10*			No	LM	λ_0 = 1.0 10/10
ALT1	1	1	4	4	10	13			No	LM	$\xi = 10^{-6}$ 10/10
ALT2	1	1	4	4	10	13			Yes	LM	$\rightarrow$ ALT1
ALT3	1	1	4	4	10	13			No	SciPy Dogbox	
ALT4	1	1	4	4	10	13			Yes	SciPy Dogbox	
ALT5	1	1	7	7	10	13	4	4	No	LM	$\rightarrow$ ALT1
ALT6	1	1	7	7	10	13	4	4	Yes	LM	$\rightarrow$ ALT1
ALT7	1	1	7	7	10	13	4	4	No	SciPy Dogbox	
ALT8	1	1	7	7	10	13	4	4	Yes	SciPy Dogbox	
ALT9	1	1	5	5	7	8	6	3	No	LM	$\rightarrow$ ALT1

Table 4.1: Alternative optimization approaches in the calibration algorithm

Figure 4.1 represents the variation of motor position errors over the number of calibrations for the different cases. Different lines show in each case the course of the maximum, minimum and mean occurring error over the entirety of the recorded calibration data. Here, only the error in the secondary axis is considered. The reason for this is that the motor positions to be set vary significantly more on this axis (see figure A.12) and thus tend to have larger errors than on the primary elevation axis. It is clear to see that the results before the 9th calibration do not significantly differ for the different methods, regarding the maximum and median error values. Immediately after the 9th calibration, a strong discrepancy becomes apparent with regard to the maximum error value. While the maximum error for ALT1-ALT4 is further reduced, the error for SOTA jumps from about 225 motor positions (5 mrad) to about 350 (8 mrad). The mean error for SOTA differs only very slightly from that of the other methods, which suggests that the jump is an isolated case. In general, no clear best method can be identified with regard to the mean error. With regard to the maximum error, ALT2 and ALT4, i.e. the methods in which the parameters are reset to their default values before each new geometry optimization, seem to be significantly more stable. In return, these two methods show higher minimum error values than the other methods in the meantime. The ranges in which the errors are located are therefore smaller for this methods.

Overall, using these four methods, after optimization with all 21 calibration points, there remains an average error around 125 motor positions (3 mrad) and a maximum of 250 motor

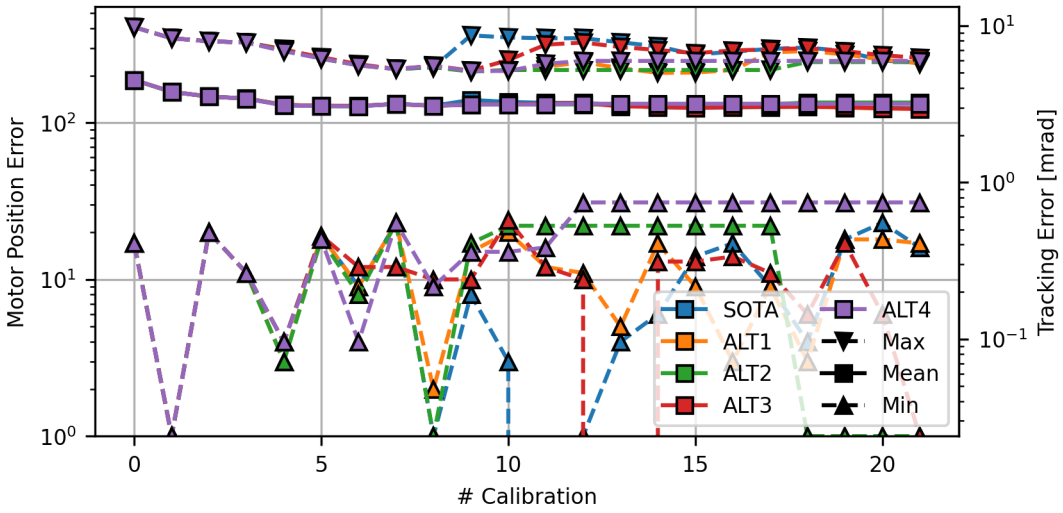


Figure 4.1: Course of the tracking error with regard to the secondary axis of Heliokon Heliostat AQ.64 in the calibration cycle for alternative approaches ALT1-ALT4 compared to the state-of-the-art (SOTA) approach

positions (6 mrad), which is not necessarily satisfactory. From this it can be concluded that the previous selection of fitting parameters should be reconsidered. As noted in section 3.1, variations in the parameters b_1 and b_2 cause non-constant angular errors that vary greatly with the range of travel of the two linear actuators. From this, four additional calibration methods ALT5-ALT8 (see table 4.1) are defined, which integrate these two parameters into the optimization from the 4th calibration point. The parameters α and β are therefore considered later, from the 7th calibration point, in the optimization. The four methods differ again only in the selection of the optimization algorithm and the resetting of the parameters. The results in figure 4.2 show a significant improvement in tracking accuracies using the four alternative methods compared to the one previously used. Optimizing b_1 and b_2 from the 4th calibration results in an immediate reduction of the tracking error of at least around 60% in the mean and maximum error compared to the SOTA algorithm. After the calibration with all 21 calibration points the tracking errors are even reduced more than 90% in comparison to the previous algorithm to a remaining mean error of 7 motor positions (0.17 mrad) and maximum error of 21 motor positions (0.5 mrad). Looking at the results of approaches ALT5-ALT8, it can be seen that from the 7th calibration point, the same results are obtained for each method. Only after the 4th calibration significant discrepancies can be seen and the methods with the SciPy Dogbox algorithm deliver the lowest mean and maximum error values. Already after the next calibration, however, the differences are no longer significant. Furthermore, it is noticeable that the error values remain almost unchanged after the 7th calibration. Since

the choice of the optimization method does not seem to have a significant influence on the results, it does not worth further investigation and ALT5 is set as the new benchmark in the following.

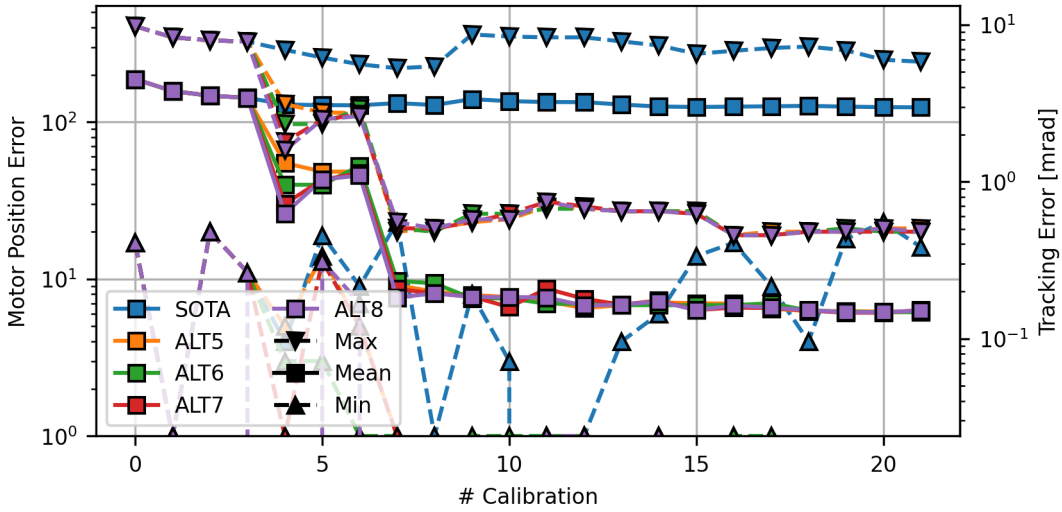


Figure 4.2: Course of the tracking error with regard to the secondary axis of Heliokon Heliostat AQ.64 in the calibration cycle for alternative approaches ALT5-ALT8 compared to the state-of-the-art (SOTA) approach

Finally, alternative method ALT9 is defined. With this method it is tried to reduce the tracking error earlier. For that b_2 is already optimized from the third calibration point, followed by α and β from the fifth point, and b_1 from the sixth. The separation of b_1 and b_2 is chosen because the effect of a faulty value of b_2 is increasing over the day, while the effect of b_1 is more constant and therefore can already be balanced by variation of the constant offset θ_K (see section 3.1). Furthermore γ is included from the 7th and δ from the 8th calibration point. From figure 4.3 it can be stated that after the calibration with 5 calibration points the mean error with ALT9 already reaches a value of around 10 motor position (0.24 mrad) which is 78% less than the mean error of ALT5. From the 7th calibration on, the differences between the two methods are negligible. This result gives reason to believe that ALT9 is the best method for calibrating Heliokon heliostats.

Since only the error on the secondary axis is considered so far, methods ALT5 and ALT9 are also compared to each other and the SOTA algorithm for the primary axis in figure A.13. Here it is also clear that ALT5 and ALT9 give much better results than SOTA and the remaining mean error after 21 calibrations is reduced from 80 (1.92 mrad) to 8 motor positions (0.192 mrad) i.e. by 90%. Furthermore, ALT9 delivers significantly better results than ALT5 and SOTA already after 5 calibrations. Also the same investigations are made with another

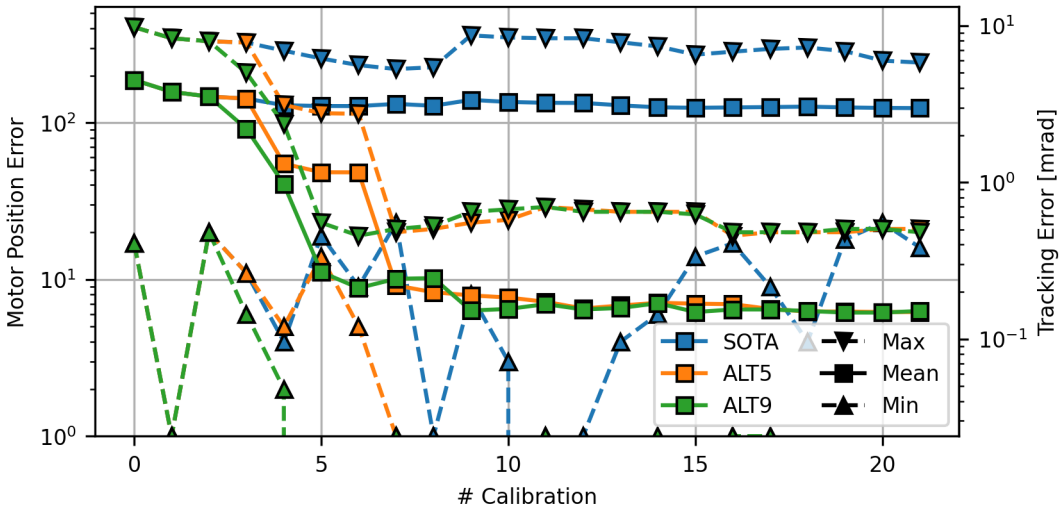


Figure 4.3: Course of the tracking error with regard to the secondary axis of Heliokon Heliostat AQ.64 in the calibration cycle for alternative approach ALT9 compared to the state-of-the-art (SOTA) and ALT5 approach

Heliostat AQ.65 and the results can be found in the appendix (see figure A.14). With this heliostat, the mean error on the secondary axis is already reduced by 70% and the maximum error by 60% compared to SOTA from the 5th calibration using ALT9. After 21 calibration points, the mean error is still reduced by 70%, while the maximum error is reduced by 75%.

5 Conclusion and Outlook

In Chapter 3, a sensitivity analysis of various geometry parameters first established that the optimization parameters selected so far are not sufficient to compensate, for example, for incorrect length parameters b_1 and b_2 on the linear actuators, which cause increasingly large angular errors as the extension length increases. Furthermore, the convergence behavior of the optimization parameters during the calibration cycle was investigated. For this purpose, a geometry model with given parameters was perturbed in individual parameters, which were corrected using calibration data generated from the exact geometry model and the existing regression algorithm. In particular, it was found that resetting geometry parameters to default values before each new optimization can have a significant impact on the convergence speed. In addition, the currently implemented Levenberg-Marquardt algorithm, which is used to regress the geometry model on recorded calibration data, was examined and significant tuning measures were tested. It has been shown that potential improvement measures to the given algorithm do not have a positive effect on the results and even worsen them significantly in some cases. Furthermore, regression algorithms from the SciPy library were tested and compared with the Levenberg-Marquardt algorithm. The SciPy algorithms were able to clearly beat the Levenberg-Marquardt algorithm in terms of the number of calibration steps required until parameter convergence.

With the knowledge gained in chapter 3, different methods were defined in chapter 4 and tested on real calibration data of the Heliokon heliostats. After several tests it was found that the selection of the optimization method does not have a significant effect on the remaining tracking errors and also the resetting of the optimization parameters before each new optimization does not bring any advantages, as it was shown in the simulated cases. Finally, it could be discovered that the optimization of the parameters b_1 and b_2 brings a significant reduction of the tracking error. Thus, in the test cases, a reduction of the remaining tracking error of up to 90 % could be determined compared to the previous method and a tracking error of less than 1 mrad could be achieved after only 5 calibrations.

After implementing one of the demonstrated methods in the heliostat field control software, it is essential to evaluate the long-term effect of the change, since the calibration data available so far is only from a very short period of time over 2 weeks. Experience has shown that heliostat tracking will deteriorate again over time, since the heliostat will never behave

consistently. Accordingly, a methodology should be developed according to which calibration data that are too old are no longer included in the model regression or are weighted less heavily. At the DLR another approach for the calibration with the help of neural networks [15] has already been investigated, where the geometry model from the Levenberg-Marquardt algorithm can be used for the pre-training of a neural network and thus the new findings from this work form a better initialization of the neural network.

Bibliography

[1] *Klima und Klimatabelle für Almeria*. https://www.klimatabelle.de/klima/europa/spanien/klimatabelle-almeria.htm. – Last visited: 2021-05-18

[2] *Klima und Klimatabelle für Köln*. https://www.klimatabelle.de/klima/europa/deutschland/klimatabelle-koeln.htm. – Last visited: 2021-05-18

[3] *Least-Squares (Model Fitting) Algorithms*. https://de.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html. – Last visited: 2021-04-13

[4] *Punktfokussierende Systeme - Die konzentrierte Kraft der Sonne*. https://www.dlr.de/sf/de/desktopdefault.aspx/tabid-7089/17123_read-43304/. – Last visited: 2021-05-26

[5] *Solarthermische Receiver - Forschung für höchste Anforderungen*. https://www.dlr.de/sf/desktopdefault.aspx/tabid-10645/18495_read-43274/. – Last visited: 2021-05-09

[6] *Solarturm-Kraftwerk Jülich feiert zehnjähriges Jubiläum*. https://www.dlr.de/content/de/artikel/news/2019/03/20190821_solarturm-Kraftwerk-juelich-feiert-zehnjaehriges.html. – Last visited: 2021-05-26

[7] BRANCH, Mary A. ; COLEMAN, Thomas F. ; LI, Yuying: A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. In: *SIAM Journal on Scientific Computing* 21 (1999), Nr. 1, S. 1–23

[8] BURISCH, Michael ; SANCHEZ, Marcelino ; OLARRA, Aitor ; VILLASANTE, Cristobal: Heliostat calibration using attached cameras and artificial targets. In: *AIP Conference Proceedings* Bd. 1734 AIP Publishing LLC, 2016, S. 130005

[9] HAGENKAMP, Markus: *Trackingverbesserung eines Heliostatenfeldes durch Optimierung von Kalibrierverfahren*, RWTH Aachen, Diplomarbeit, 2018

[10] HINKLEY, Jim ; CURTIN, Bryan ; HAYWARD, Jenny ; WONHAS, Alex ; BOYD, Rod ; GRIMA, Charles ; TADROS, Amir ; HALL, Ross ; NAICKER, Kevin ; MIKHAIL, Adeeb: *Concentrating solar power: drivers and opportunities for cost-competitive Electricity*. CSIRO Victoria, 2011

[11] HOEGH-GULDBERG, Ove ; JACOB, Daniela ; BINDI, M ; BROWN, S ; CAMILLONI, I ; DIEDHIOU, A ; DJALANTE, R ; EBI, K ; ENGELBRECHT, F ; GUIOT, J u. a.: Impacts of 1.5 C global warming on natural and human systems. In: *Global warming of 1.5 C. An IPCC Special Report* (2018)

[12] MADSEN, Kaj ; NIELSEN, Hans B. ; TINGLEFF, Ole: Methods for non-linear least squares problems. (2004)

[13] MORÉ, Jorge J.: The Levenberg-Marquardt algorithm: implementation and theory. In: *Numerical analysis*. Springer, 1978, S. 105–116

[14] MORÉ, Jorge J. ; GARBOW, Burton S. ; HILLSTROM, Kenneth E.: User guide for MINPACK-1 / CM-P00068642. 1980. – Forschungsbericht

[15] PARGMANN, Max ; QUINTO, Daniel M. ; SCHWARZBÖZL, Peter ; PITZ-PAAL, Robert: High accuracy data-driven heliostat calibration and state prediction with pretrained deep neural networks. In: *Solar Energy* 218 (2021), S. 48–56

[16] QUERO, M ; KORZYNIETZ, Roman ; EBERT, Miriam ; JIMÉNEZ, AA ; DEL RÍO, A ; BRIOSO, JA: Solugas–Operation experience of the first solar hybrid gas turbine system at MW scale. In: *Energy Procedia* 49 (2014), S. 1820–1830

[17] ROEB, Martin ; SÄCK, J-P ; RIETBROCK, Prahl ; PRAHL, Christoph ; SCHREIBER, Heike ; NEISES, Martina ; OLIVEIRA, Lamark de ; GRAF, D ; EBERT, M ; REINALTER, W u. a.: Test operation of a 100 kW pilot plant for solar hydrogen production from water on a solar tower. In: *Solar Energy* 85 (2011), Nr. 4, S. 634–644

[18] STIEGLITZ, Robert ; HEINZEL, Volker: *Thermische Solarenergie: Grundlagen, Technologie, Anwendungen*. Springer-Verlag, 2013

[19] STONE, Kenneth W.: *Automatic heliostat track alignment method*. Januar 14 1986. – US Patent 4,564,275

[20] TRANSTRUM, Mark K. ; SETHNA, James P.: Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization. In: *arXiv preprint arXiv:1201.5885* (2012)

[21] ULLRICH, Sven: *DLR präsentiert solaren Wasserstoff-freaktor*. `https://www.erneuerbareenergien.de/archiv/dlr-praesentiert-solaren-wasserstoffreaktor-150-436-105617.html`. – Last visited: 2021-05-08

[22] VOGLIS, C ; LAGARIS, IE: A rectangular trust region dogleg approach for unconstrained and bound constrained nonlinear optimization. In: *WSEAS International Conference on Applied Mathematics* Bd. 7 Citeseer, 2004

Appendix

$$\overline{\overline{R}}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{pmatrix} \tag{A.1a}$$

$$\overline{\overline{R}}_y(\phi) = \begin{pmatrix} cos(\phi) & 0 & -sin(\phi) \\ 0 & 1 & 0 \\ sin(\phi) & 0 & cos(\phi) \end{pmatrix} \tag{A.1b}$$

$$\overline{\overline{R}}_z(\phi) = \begin{pmatrix} cos(\phi) & sin(\phi) & 0 \\ -sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{A.1c}$$

$$atan2(x,y) = \begin{cases} arctan(\frac{y}{x}) & x > 0 \\ \frac{\pi}{2} - arctan(\frac{x}{y}) & y > 0 \\ -\frac{\pi}{2} - arctan(\frac{x}{y}) & y < 0 \\ \pm\pi & x < 0 \\ undefined & else \end{cases} \tag{A.2}$$

```
def lm(f,x0):
    mu = 1.0; gtol = 1e-12; ftol = 1e-4; kmax = 500
    x = x0; k = 0
    init_loss = 0.5 * transpose(f(x)) @ f(x)
    J = jacobian(f,x)
    Jt = transpose(J)
    H = Jt @ J
    g = Jt @ f
    compute_derivatives = False
    found = norm(g,inf) <= gtol
```

```
    while not found and k <= kmax:
        k = k + 1
        h = solve(H + mu*I,-g)
        xnew = x + h
        new_loss = 0.5 * transpose(f(xnew)) @ f(xnew)
        if new_loss < init_loss:
            x = xnew
            found = init_loss - new_loss <= ftol*init_loss
            init_loss = new_loss
            compute_derivatives = True
            mu = mu/10.0
        else:
            mu = mu*10.0
        if not found and compute_derivatives:
            J = jacobian(f,x)
            Jt = transpose(J)
            H = Jt @ J
            g = Jt @ f
            found = norm(g,inf) <= gtol
            compute_derivatives = False
    return x
```

Algorithm A.1: State-of-the-art Levenberg-Marquardt algorithm

Seed	θ_K	τ_K	α	β	γ	δ
0	0.00852	1.05267	1.6246	1.5942	-0.002665	0.005093
1	-0.1448	1.05536	1.3091	1.4673	-0.01233	-0.01423
2	-0.01117	0.6918	1.5968	1.5369	-0.00278	-0.005922
3	0.008866	1.04899	1.4613	1.5764	0.01372	0.01383
4	0.08151	0.9647	1.8183	1.6833	0.006902	-0.00991

Table A.1: Randomly generated parameter values (in rad) within their tolerance limits using different seeds

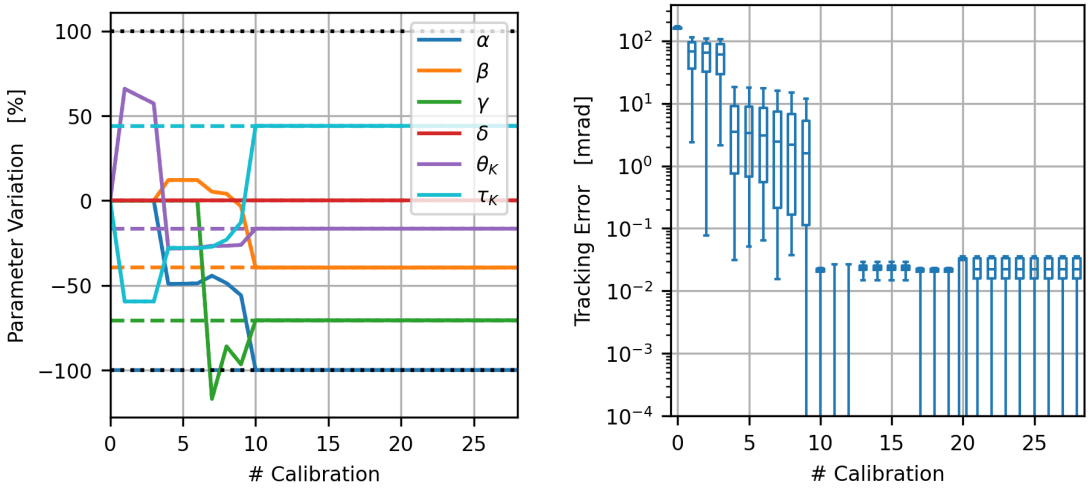


Figure A.1: Calibration results with initially faulty parameters θ_K, τ_K, α, β and γ, and resetting parameters

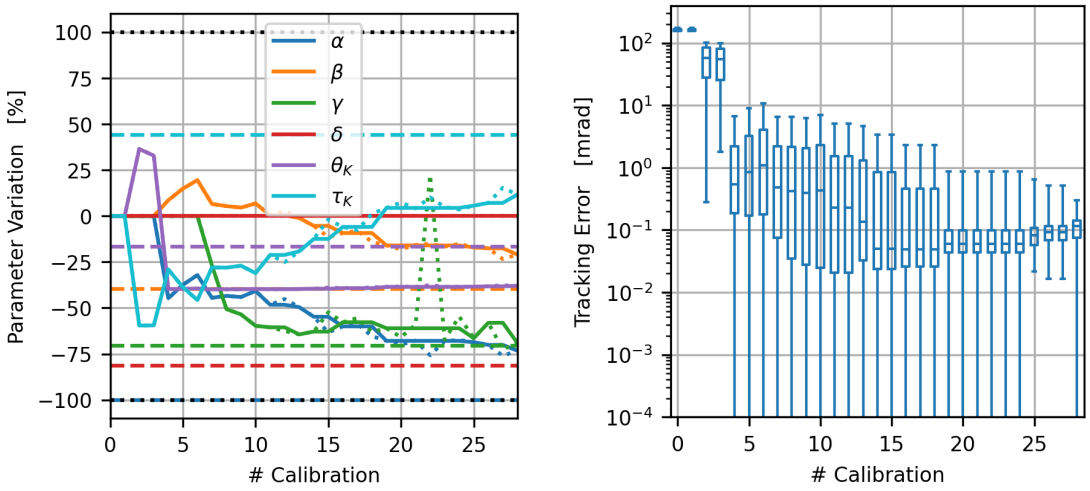


Figure A.2: Calibration results with completely initially faulty optimization parameters and resetting parmaters

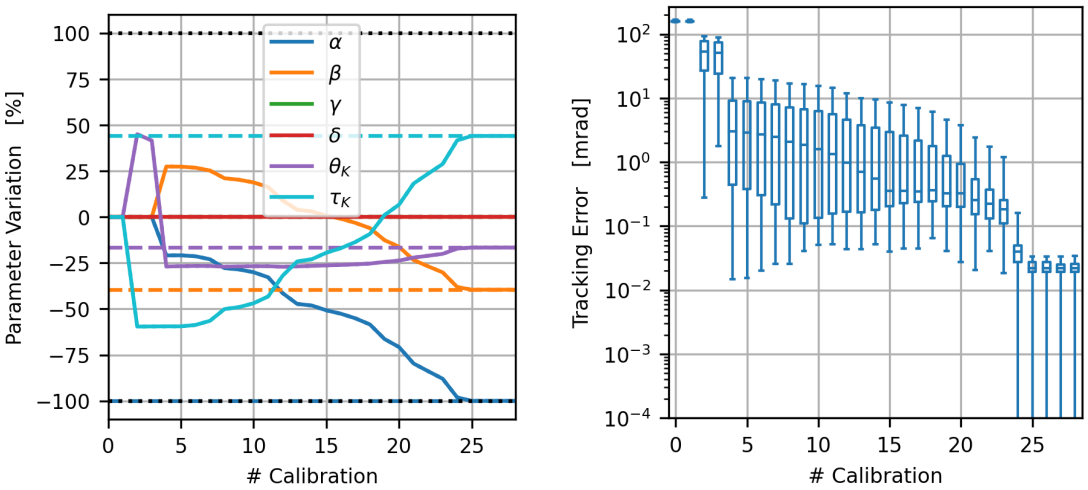


Figure A.3: Calibration results with only optimizing initially faulty parameters θ_K, τ_K, α and β, without resetting parameters

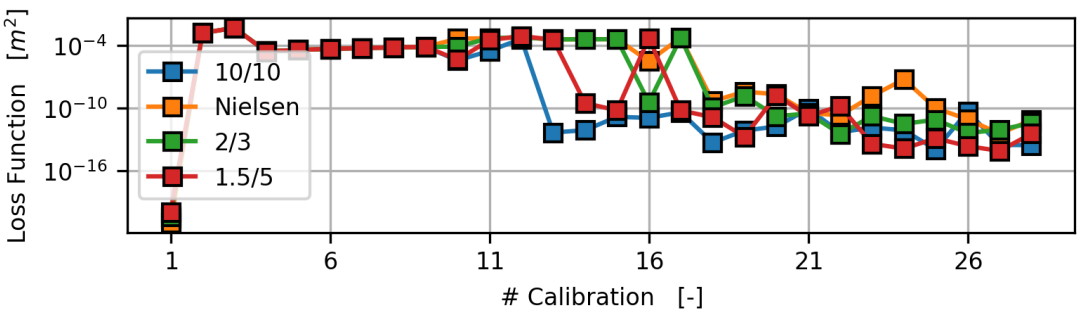


Figure A.4: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied update strategy of the damping parameter λ for real geometry parameter values generated with random generator seed 0

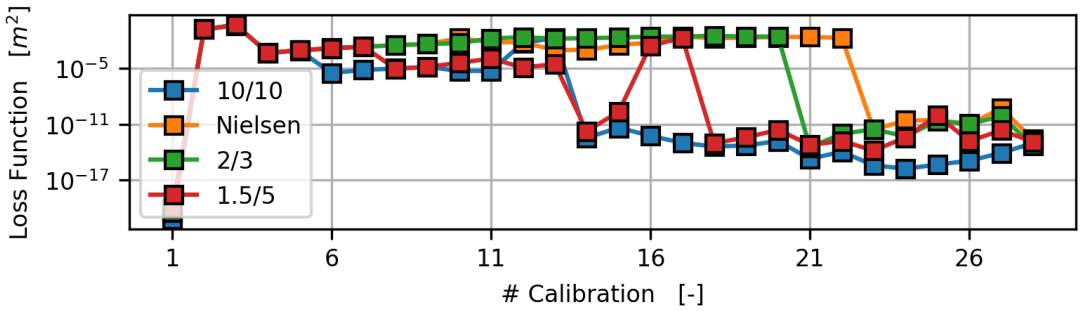


Figure A.5: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied update strategy of the damping parameter λ for real geometry parameter values generated with random generator seed 2

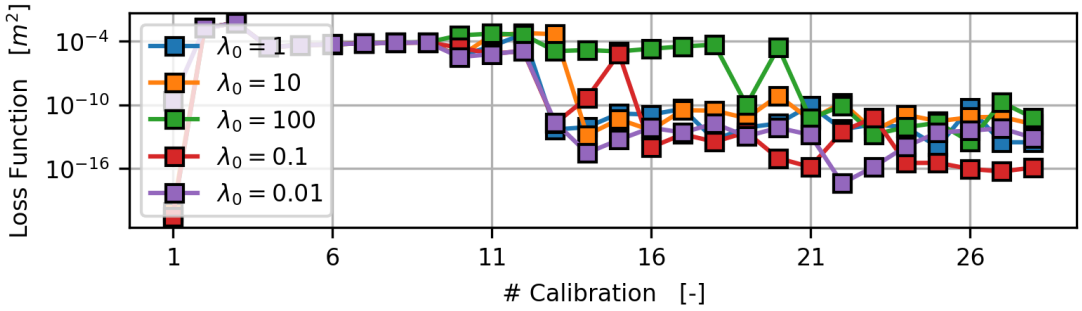


Figure A.6: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied initial value of the damping parameter λ for real geometry parameter values generated with random generator seed 0

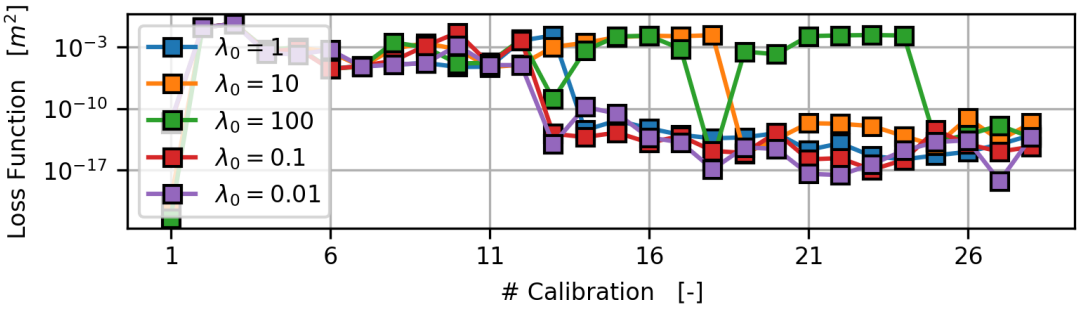


Figure A.7: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied initial value of the damping parameter λ for real geometry parameter values generated with random generator seed 2

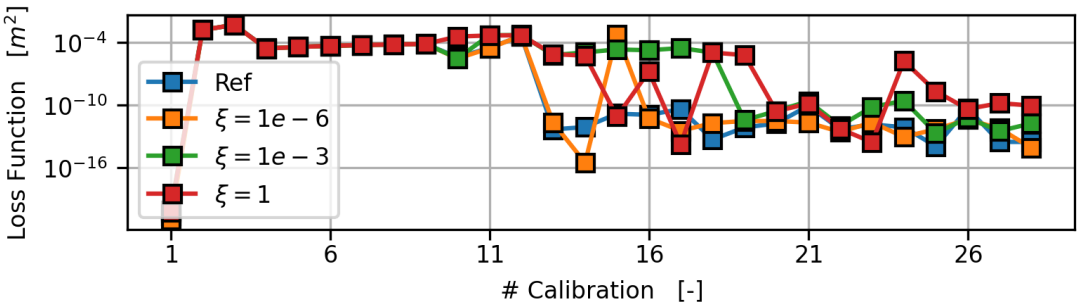


Figure A.8: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied factor ξ relating the initial damping parameter λ to the initial Jacobian values of the target function for real geometry parameter values generated with random generator seed 2

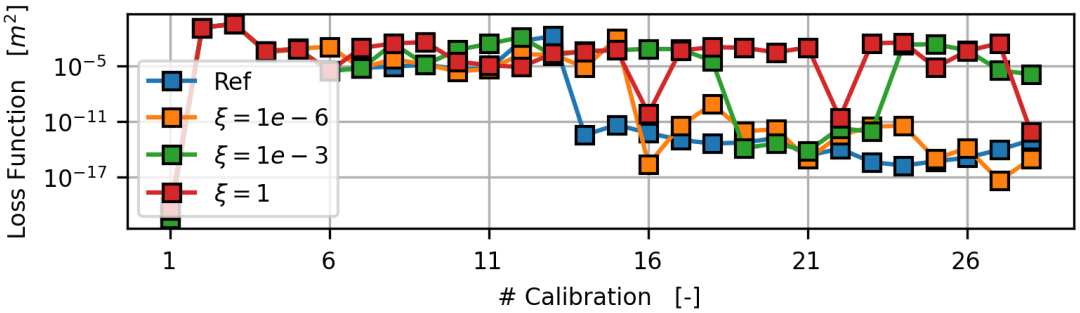


Figure A.9: Levenberg-Marquardt algorithm performance for increasing number of calibration data and varied factor ξ relating the initial damping parameter λ to the initial Jacobian values of the target function for real geometry parameter values generated with random generator seed 0

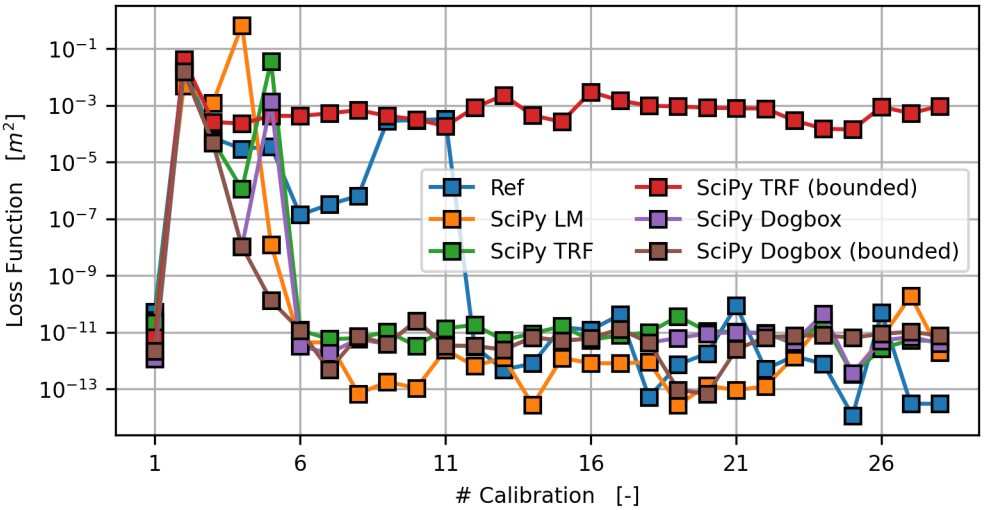


Figure A.10: Comparison of the different SciPy least-squares problem solver with the current Levenberg-Marquardt algorithm

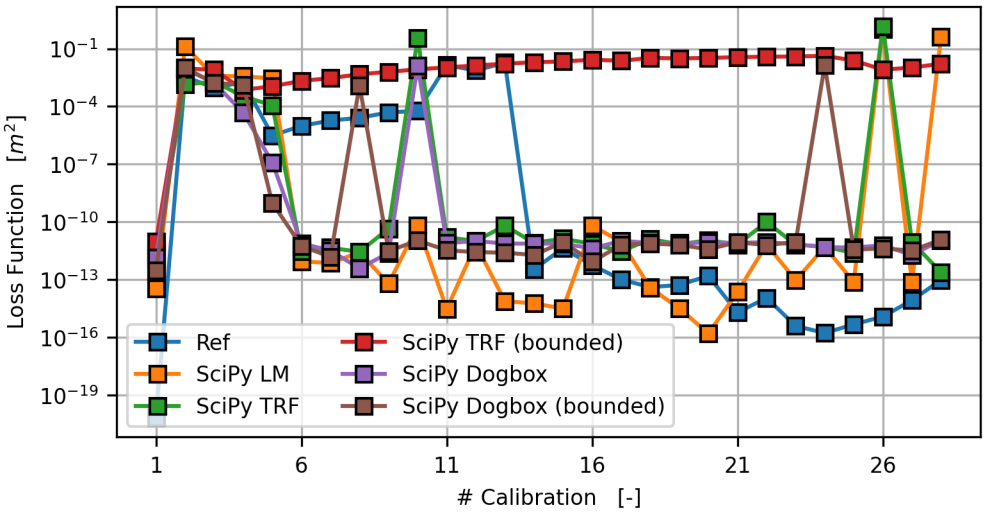


Figure A.11: Comparison of the different SciPy least-squares problem solver with the current Levenberg-Marquardt algorithm

Parameter	Initial Value	Unit
h_e	123.16	m
h_n	100.72	m
h_u	88.79	m
θ_K	0.0	rad
τ_K	0.94	rad
α	1.571	rad
β	1.571	rad
γ	0.0	rad
δ	0.0	rad
v_{12}	0.0	m
v_{2M}	0.1755	m
a_1	0.0	m
b_1	0.07	m
c_1	0.335308	m
d_1	0.338095	m
e_1	0.0	m
a_2	0.0	m
b_2	0.07	m
c_2	0.340771	m
d_2	0.3191	m
e_2	0.0	m

Table A.2: Geometry parameter values and tolerances of heliostat AQ.64 from heliostat field in Jülich

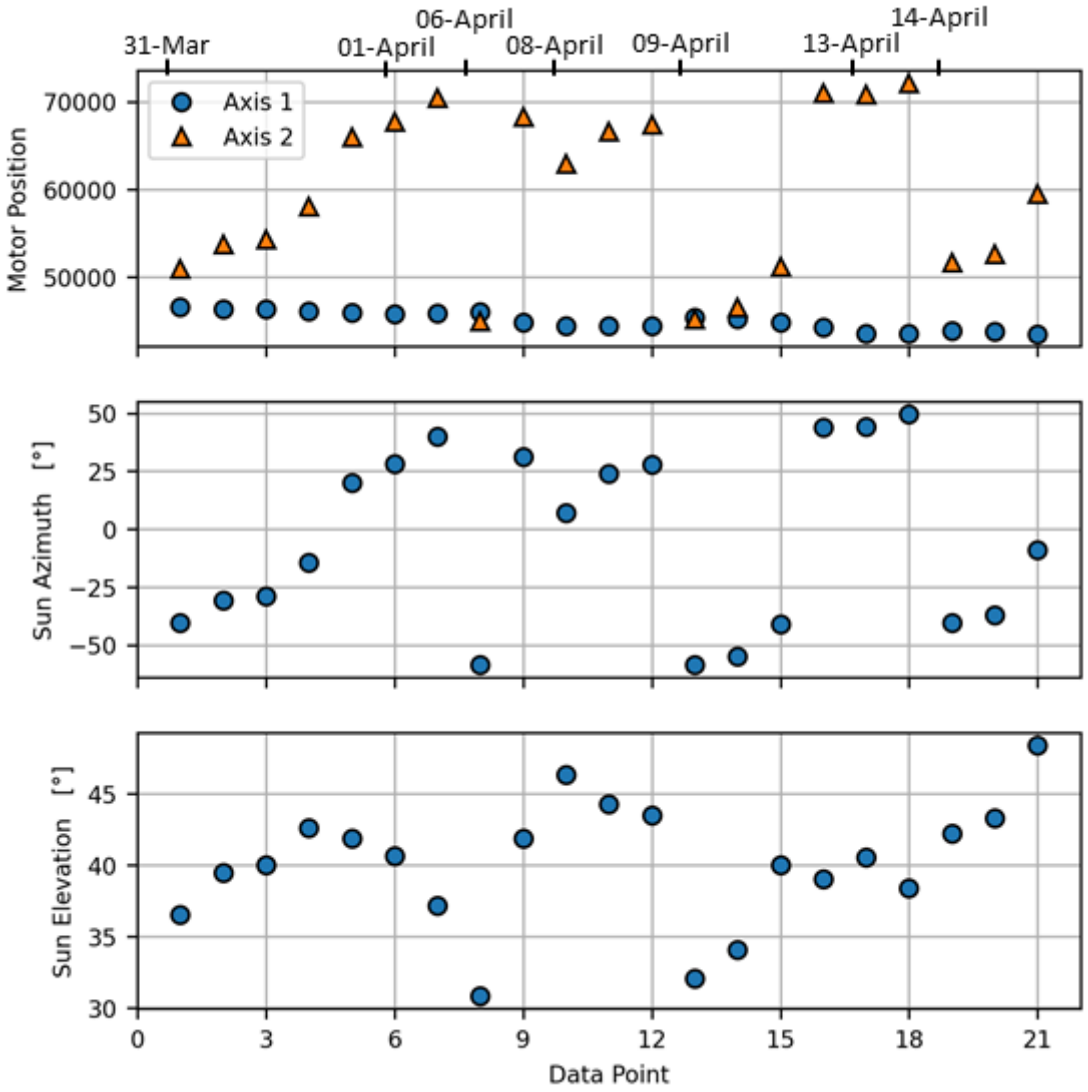


Figure A.12: Calibration data of *Heliokon* Heliostat AQ.64, collected between 31.03.2021 and 14.04.2021 at the solar tower power plant in Jülich

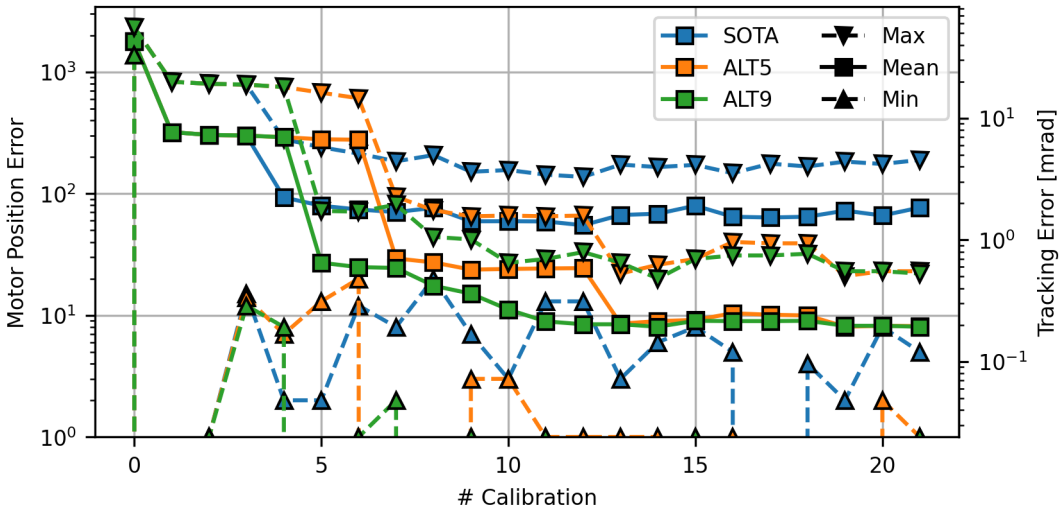
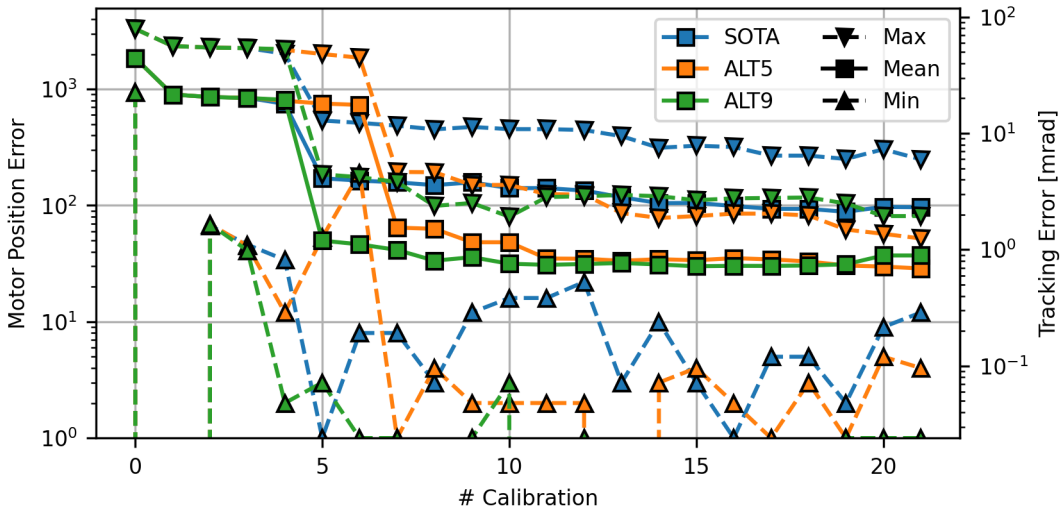
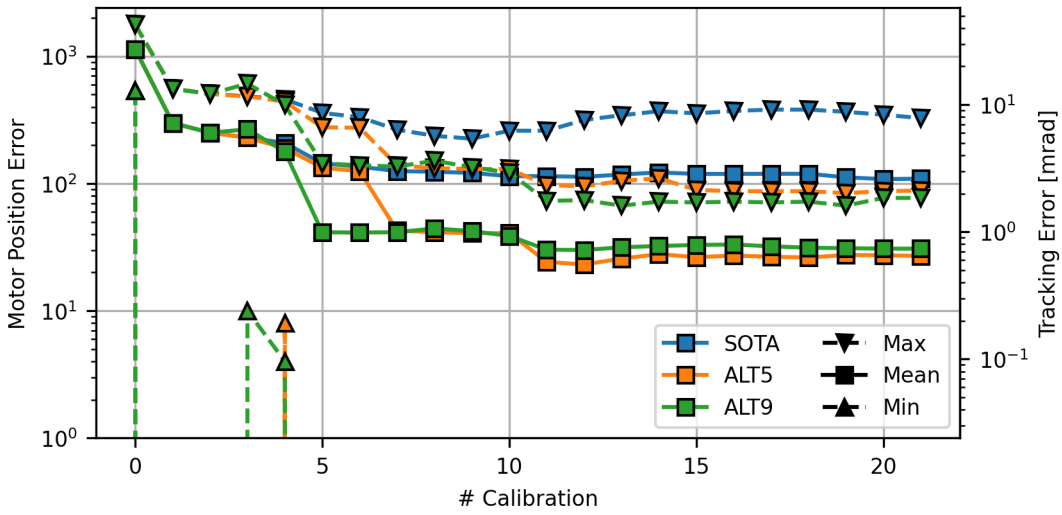


Figure A.13: Course of the tracking error with regard to the primary axis of Heliokon Heliostat AQ.64 in the calibration cycle for alternative approach ALT9 compared to the state-of-the-art (SOTA) and ALT9 approach



(a) Primary axis



(b) Secondary axis

Figure A.14: Course of the tracking error of Heliokon Heliostat AQ.65 in the calibration cycle for alternative approach ALT9 compared to the state-of-the-art (SOTA) and ALT5 approach